



ユーザーガイド

AWS 変換



AWS 変換: ユーザーガイド

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Transform とは AWS	1
用語	1
開始方法	4
設定	4
[開始する前に]	4
にサインアップする AWS アカウント	4
IAM 専用アクセスの開始方法	4
IAM 認証情報の使用	5
の開始方法 AWS Organizations	6
の開始方法 AWS IAM アイデンティティセンター	7
サードパーティーの ID プロバイダーの使用	8
オンボーディング	11
独自の S3 バケットの使用	12
AWS 変換を有効にする	21
クイックスタート: AWS 変換の試行	23
ユーザーの管理	23
IAM アイデンティティセンターへのユーザーの追加	23
AWS 変換へのユーザーの追加	24
IAM のみのアクセスを持つユーザーの管理	25
共同作業者のアクセス許可について	25
AWS 変換環境	28
言語設定	29
プロジェクトを開始する	30
AWS チャットを変換する	31
デベロッパーツール	33
Kiro Power	33
エージェントプラグイン	33
IDE プラグイン	34
MCP サーバー	34
認証	35
AWS 変換用のカスタムエージェントを構築する	35
移行評価	37
前提条件	37
移行評価の仕組み	37

インベントリデータのアップロード	38
検出結果の確認	39
インベントリスコープの管理	39
評価シナリオの設定	39
評価機能	40
Amazon EC2 の適切なサイズ設定	40
Amazon EBS ストレージ	40
Amazon FSx	40
Amazon RDS for SQL Server	40
Amazon EC2 SQL Server	41
オンプレミス料金	41
持続可能性	42
ビジネス価値評価	42
ネットワークコスト	42
サポートコスト	42
エンドユーザーコンピューティング	43
チャットベースの評価の使用	43
限られたデータによる大まかな推定	43
チャットによるインベントリの追加	43
コストの変更とサービスの追加	43
シナリオの比較	44
シナリオの作成	44
比較の実行	44
What-if 分析	45
成果物の生成	45
関連トピック	45
カスタム	46
AWS Transform custom とは	46
主な機能	46
変換パターン	46
AWS Transform カスタムの仕組み	49
の重要な概念の理解	49
変換の定義	49
変換レジストリ	50
ドラフト変換と公開変換	50
リファレンスと教訓	51

ビルドおよび検証コマンド	51
継続的な学習	52
クライアント側のスキル	53
開始方法	53
前提条件	53
Transform CLI AWS のインストール	54
認証の設定	55
AWS リージョンの設定	57
最初の変換の実行	60
変換結果について	61
カスタム変換の作成	61
AWS ウェブアプリケーションの変換 (オプション)	62
カスタム変換コマンドの概要	62
ワークフロー	63
変換の実行	63
継続的な学習	68
高度な設定	69
カスタム変換を作成する	83
コマンドリファレンス	87
インタラクティブコマンド	87
変換定義コマンド	87
Lesson コマンド	91
タグコマンド	91
コマンドの更新	91
MCP コマンド	92
AWSマネージド変換	92
概要	92
使用可能な AWSマネージド変換	92
AWSマネージド変換のカスタマイズ	108
一般的なユースケース	109
Lambda ランタイムのアップグレード	109
VPC エンドポイント (AWS PrivateLink)	110
AWS Transform カスタム VPC エンドポイントに関する考慮事項	111
前提条件	111
AWS Transform カスタムのインターフェイス VPC エンドポイントの作成	112
AWS Transform カスタムの VPC エンドポイントポリシーの作成	112

オンプレミスコンピュータを使用して AWS Transform カスタムエンドポイントに接続する	113
トラブルシューティング	113
ログの場所	113
一般的な問題	114
サポート情報	116
S3 アクセスの問題	116
CLI バージョン履歴	118
継続的なモダナイゼーション	127
AWS Transform Continuous Modernization とは	127
主な機能	127
分析タイプ	128
重要な概念の理解	129
[Sources] (出典)	129
リポジトリ	129
分析	130
検出結果	130
修復	130
変換定義	130
AWS Transform の継続的なモダナイゼーションの仕組み	130
コンピューティングオプション	131
セキュリティエージェントの設定	133
開始方法	134
前提条件	134
クイックスタート	134
継続的なモダナイゼーションの使用	135
ソース管理	135
リポジトリの検出と管理	136
分析の実行	137
検出結果の管理	137
修復の作成	138
AWS ウェブアプリケーションの変換 (オプション)	139
デベロッパーツール	140
エージェントスキル	140
CLI リファレンス	141
トラブルシューティング	143

検出ツール	145
ワークフロー	145
検出ツールをセットアップする	146
前提条件	146
VMware にデプロイする	148
Hyper-V にデプロイする	149
Linux にデプロイする	150
検出ツールを設定する	153
必要なアクセス許可	165
VMware vCenter	165
Hyper-V ホスト (Windows)	166
Linux サーバー (SSH)	166
Windows サーバー (WinRM) — OS メトリクス	168
Windows サーバー (WinRM) — SQL Server コレクション	169
Oracle データベース (SQL)	170
ネットワークコレクション	171
クイックリファレンス: ユースケース別の最小アクセス許可	171
データ収集	172
収集スケジュール	172
検出されたインベントリ	174
OS 関連データ	178
セキュリティに関する考慮事項	195
検出ツール VM の保護	195
認証情報の保護	195
認証情報ストレージ	197
注意して自動接続機能を使用する	197
CSV インポートのセキュリティ	198
アクセスの取り消しに関する考慮事項	199
検出ツールのトラブルシューティング	199
vCenter への検出ツールの接続の検証	199
WinRM のトラブルシューティング	201
Kerberos のトラブルシューティング	201
Oracle Database のトラブルシューティング	206
SNMP のトラブルシューティング	207
ネットワーク収集エラー	208
OS メトリクス収集エラー	209

検出されたインベントリのアクセスの問題	209
SSH キー認証のトラブルシューティング	210
Linux インストーラのトラブルシューティング	211
一般的なエラーメッセージ	211
リリースノート	212
Connector	217
エージェントオペレーションのユーザートレーサビリティ	217
コネクタの管理	219
移行 (VMware を含む)	221
機能と主要機能	221
アーキテクチャ	221
移行ジョブ	222
ワークスペースの取得	222
ジョブタイプ	223
ジョブの作成と開始	224
ワークスペースの概要レポートのダウンロード	224
制限事項	225
ソースデータを検出する	225
移行計画を立てる	226
ワークフロー	228
7Rs レコメンデーション	230
図とレポート	231
ターゲットの AWS アカウントとリージョンを接続する	232
ステップ 1: 移行タイプの選択	232
ステップ 2: MAP 契約	232
ステップ 3: コネクタ設定	233
ランディングゾーンを構築する	239
コネクタのセットアップ	239
基盤のセットアップ	240
ワークロードアカウント設計	244
IaC 形式	246
デプロイの承認プロセス	246
ランディングゾーンリソースにタグを付ける	247
変更の取り消し	247
関連リソース	248
ネットワークを に移行する AWS	248

ステップ 1: ソースネットワークマッピング	249
ステップ 2: 追加の設定ファイル	250
ステップ 3: ネットワークトポロジ	250
ステップ 4: セキュリティグループのマッピング	254
ステップ 5: ネットワークを確認して最適化する	256
ステップ 6: ネットワーク図	260
ステップ 7: リソースのタグ付けを設定する	261
ステップ 8: ネットワークをデプロイする	262
デプロイされたネットワークリソースを削除する	264
設定ファイルの抽出	264
サーバーの移行	265
前提条件と移行デフォルトの設定	266
ステップ 1: 移行ウェブを設定する	269
ステップ 2: インベントリを検証して確認する	272
ステップ 3: レプリケーションエージェントをデプロイする	273
ステップ 4: データレプリケーション	280
ステップ 5: テスト	281
ステップ 5b: アプリケーションをカットオーバー準備完了としてマークする	282
ステップ 6: カットオーバー	282
サーバーライフサイクルの状態	283
デプロイの承認	284
リリースノート	284
2026 年 6 月	284
2026 年 5 月	285
2026 年 4 月	285
2026 年 3 月	285
2026 年 2 月	286
2026 年 1 月	286
2025 年 12 月	286
2025 年 10 月	286
2025 年 9 月	286
2025 年 8 月	287
2025 年 5 月	287
ソースコードのコンテナ化	288
機能と主要機能	288
コンテナ化の仕組み	289

前提条件	289
コンテナ化ワークフロー	290
コンテナ化ジョブの開始	290
アクセス許可	291
基本ポリシー	291
ネットワークポリシー	292
ストレージポリシー	292
AWS KMS ポリシー	292
Amazon ECS ポリシー	292
Amazon EKS ポリシー	293
ステップ 1: セキュリティに関する免責事項	293
免責事項の内容	293
必要な作業	294
ステップ 2: ソースコードのクローンを作成する	294
オプション 1: Git リポジトリ	294
オプション 2: Zip ファイルのアップロード	295
オプション: プライベート依存関係を設定する	296
コンテナ化設定の確認	296
ステップ 3: コンテナ化	297
コンテナ化中に何が起こるか	297
サポートされるアプリケーションタイプ	297
必要な作業	298
ステップ 4: アーティファクトを確認する	298
Git リポジトリからの変更の確認	298
zip アップロードからの変更の確認	298
変更のリクエスト	299
ステップ 5: イメージを公開する	299
公開中に何が起こるか	299
必要な作業	299
ステップ 6: IaC を生成する	300
Amazon Elastic Kubernetes Service (Kubernetes)	300
Amazon Elastic Container Service (Terraform)	301
自動検証とセキュリティスキャン	301
公開されたイメージなしで IaC を生成する	301
必要な作業	302
ステップ 7: テストインフラストラクチャをデプロイする	302

テストデプロイ中に何が起こるか	302
必要な作業	302
インフラストラクチャでの反復処理	303
セッション復旧	303
ステップ 8: テストインフラストラクチャをクリーンアップする	303
必要な作業	303
ステップ 9: カットオーバーインフラストラクチャをデプロイする	304
必要な作業	304
セッション復旧	305
Mainframe Modernization	306
機能と主要機能	306
大まかなウォークスルー	307
ヒューマンインザループ (HITL)	307
メインフレームアプリケーションの変換でサポートされるファイルタイプ	308
AWS Transform メインフレームでサポートされているリージョンとクォータ	309
モダナイゼーションジャーニー	309
フェーズ 1: 評価	309
フェーズ 2: 動員	310
フェーズ 3: 移行とモダナイズ	311
フェーズ 4: 運用、最適化、イノベーション	313
メインフレームアプリケーションワークフローの変換	314
前提条件: S3 でプロジェクト入力を準備する	314
メインフレームソースアーティファクトコレクション	315
サインインしてジョブを作成する	324
変換の進行状況の追跡	325
コネクタのセットアップ	325
ジョブプランの評価と再考	330
ジョブプランを再考する	334
カスタムジョブプラン	334
モダナイズされたアプリケーションを構築してデプロイする	365
前提条件	366
ステップ 1: モダナイズされたコードを取得する	366
ステップ 2: モダナイズされたアプリケーションを構築する	366
ステップ 3: テスト環境を設定する	367
ステップ 4: モダナイズされたアプリケーションをデプロイする	368
ステップ 5: モダナイズされたアプリケーションをテストする	369

その他の例	369
フルスタックの Windows モダナイゼーション	370
.NET	370
エクスペリエンス	370
機能と主要機能	371
サポートされているバージョンとプロジェクトタイプ	371
制限事項	372
人間による介入	372
詳細情報	373
.NET ウェブアプリ	373
.NET IDE	389
UI レイヤーコードの移植	397
変換レポート	399
ベストプラクティス	401
SQL Server のモダナイゼーション	407
サポート対象のリージョン	407
機能と主要機能	407
サポートされているバージョンとプロジェクトタイプ	408
SQL Server の要件	413
アプリケーション要件	413
インフラストラクチャ要件	414
リポジトリの要件	415
SQL Server のセットアップ	415
AWS 変換設定	423
SQL Server モダナイゼーションジョブを作成する	423
SQL Server モダナイゼーションワークフロー	431
変換後のテストとデプロイ	462
ベストプラクティスとトラブルシューティング	464
セキュリティ	470
データ保護	471
クロスリージョン処理	472
データ暗号化	481
サービスの改善	491
ID とアクセス管理	492
オーディエンス	493
アイデンティティを使用した認証	493

ポリシーを使用したアクセスの管理	494
AWS Transform と IAM の連携方法	496
アイデンティティベースのポリシーの例	502
トラブルシューティング	511
サービスにリンクされたロールの使用	513
AWS 管理ポリシー	519
AWS 変換アクセス許可リファレンス	535
コンプライアンス検証	537
耐障害性	537
VPC エンドポイント (AWS PrivateLink)	538
AWS Transform VPC エンドポイントに関する考慮事項	538
前提条件	538
のインターフェイス VPC エンドポイントの作成 AWS Transform	539
オンプレミスコンピュータを使用して AWS Transform エンドポイントに接続する	540
ウェブアプリケーション VPC アクセス	540
仕組み	540
アーキテクチャ	541
前提条件	542
VPC エンドポイントポリシーの設定	542
制御されたインターネット出力の設定	543
検証	550
トラブルシューティング	551
コストに関する考慮事項	552
クリーンアップ	552
モニタリング	554
CloudTrail ログ	554
AWS CloudTrail での情報の変換	555
AWS 変換ログファイルエントリについて	556
通知	556
E メール通知	556
クォータ	559
サポート対象のリージョン	563
Supported AWS リージョン (デフォルトで有効)	563
サポート	565
前提条件	565
サポートケース管理の有効化	565

サポートケースの作成	565
サポートケースの表示	566
ケースへの通信の追加	567
ケースの解決	567
サポートケース管理の無効化	567
使用状況テレメトリ	568
運用データ	569
変更ログ	570
ドキュメント履歴	593
.....	dcii

Transform とは AWS

AWS Transform は、エージェント AI エクスペリエンスでインフラストラクチャ、アプリケーション、コードの変換を高速化および簡素化するのに役立つサービスです。さまざまなワークロードにわたるモダナイゼーションと移行の作業を合理化するための特殊なツールを提供します。

AWS 変換を使用すると、次のことができます。

- IBM z/OS と富士通 GS21 をモダナイズして に移行する AWS
- VMware ワークロードを Amazon EC2 に移行する
- .NET アプリケーションを Linux 対応のクロスプラットフォーム .NET にモダナイズする
- 移行の準備状況に関するワークロードの評価

AWS トランスフォームは、検出、計画、実行の各フェーズで、手間のかかる複雑なトランスフォーメーションタスクを、言語、フレームワーク、インフラストラクチャに関する専門知識を持つ AI エージェントにオフロードするのに役立ちます。このアプローチにより、チームは統合ウェブエクスペリエンスを通じて複雑で大規模なプロジェクトを効率的に変革しながら、イノベーションに集中できます。

Transform の使用には追加料金はかかりません AWS。

用語

このセクションでは、斜体は別の用語の定義内の公式用語を示します。

アカウント接続

このコンテキストのアカウントは、または GitHub アカウントなど、AWS またはリモートサービスのリソースに対する、顧客所有のコンテナ AWS アカウント またはセキュリティ境界への一般的な参照です。

アーティファクト

Transform によって生成される出力成果物 AWS。

管理者

管理者はワークスペースのすべてを読み取り、変更することができます。AWS 変換によるチャットの開始、ジョブの開始と停止、アーティファクトのアップロード/ダウンロードを行うこ

とができます。管理者は、ヒューman-in-the-loop (HITL) アクションの実行ジョブを操作し、メインへのマージ、グラフ分解の実行、本番環境へのコードのデプロイなどの重要な HITL アクションを承認できます。管理者はワークスペース[???](#)、[コネクタ](#)、ユーザーを変更できます。

エージェント

特定の変換タイプを実行するタスク固有のサービス。たとえば、VMware の移行などです。

承認者

承認者には管理者と同様のアクセス許可がありますが、ワークスペース、コネクタ、またはユーザーを変更するアクセス許可はありません。承認者は[ワークスペース](#)、[コネクタ](#)、またはユーザーを変更することはできません。

アセット

変換[ジョブ](#)の入力。たとえば、顧客のソースコード、サーバー、データベース、ネットワークなどです。アセットには[コネクタ](#)経由でアクセスします。

コラボレーターリクエスト

Transform AWS が人間に何かを求めるタスク。

コネクタ

コネクタは、AWS Transform の外部のシステム内の顧客所有のリソースへのアクセスを許可するアセットプロバイダーです。

コネクタを設定するときは、接続先のアカウントの管理者が接続を受け入れる必要があります。接続を受け入れるには、[コネクタアクセプタポリシー](#)で付与されたアクセス許可が必要です。

次の 2 つのアカウントは、同一であるか、同じ AWS Organizations 組織内にある必要があります。

- Transform AWS 管理者がサービスを有効にするアカウント。
- 変換の受信側になるアカウント。このアカウントには、[コネクタ](#)の使用を許可する IAM ロールを割り当てる必要があります。

寄稿者

寄稿者はワークスペースですべてを読み取ることができます。変換による AWS チャットの開始、ジョブの開始と停止、アーティファクトのアップロードとダウンロード、HITL アクションの実行[中のジョブ](#)の操作を行うことができます。ただし、メインへのマージ、グラフ分解の実行、本番環境へのコードのデプロイなど、重要な HITL アクションを実行することはできません。また、寄稿者は[ワークスペース](#)、[コネクタ](#)、またはユーザーを変更することはできません。

目標

AWS Transform が到達するように動作するユーザー定義の終了状態。ユーザーはこれを記述し、AWS Transform は必要に応じてユーザーと連携して実行する一連のタスクに目的を変換します。

ジョブ

ユーザーが定義した [目標を達成](#) するために AWS Transform が取り組んでいる長時間実行されるプロセス (週/月+)。複数の [タスク](#) と [共同作業者のリクエスト](#) で構成されます。

計画

Transform が [目的](#) を達成するために (人間のユーザーの助けを借りて) AWS 引き受ける [タスク](#) のリスト。

リーダー

読者は AWS 変換ジョブのステータスと結果を表示できますが、変更を加えることはできません。 [ワークスペース](#) 内のすべての読み取り、アーティファクトのダウンロード、 [ジョブ](#) の表示、ヒューman-in-the-loop (HITL) アクションの表示を行うことができます。ただし、読者は AWS 変換で変更アクションを実行することはできません。

タスク

[ジョブ](#) の一部である個々の作業単位。

作業ログ

Transform AWS とユーザーが [ジョブ](#) の一部として実行したアクションのログ。

Workspace

[コネクタ](#) や [ジョブ](#) などの他のリソースを含む AWS 変換リソース。 [ワークスペース](#) はアクセス許可の境界として機能します。

AWS 変換の開始方法

トピック

- [AWS 変換の設定](#)
- [AWS 変換を有効にする](#)
- [クイックスタート: AWS 変換の試行](#)
- [ユーザーの管理](#)
- [AWS 変換環境](#)

AWS 変換の設定

[開始する前に]

AWS Transform を設定する前に、管理者権限を持つ AWS アカウントがあることを確認してください。

Note

proof-of-conceptとして AWS 変換を試す場合やテスト環境の場合は、[「クイックスタート: AWS 変換を試す」](#)を参照してください。

にサインアップする AWS アカウント

の使用を開始するには AWS、が必要です AWS アカウント。の作成の詳細については AWS アカウント、「AWS アカウント管理 リファレンスガイド」の「[の開始方法 AWS アカウント](#)」を参照してください。

IAM 専用アクセスの開始方法

IAM のみのアクセスでは、ユーザーは既存の AWS 認証情報で認証されます。

IAM のみのアクセスを設定するには:

1. AWS コンソールで、AWS 変換に移動し、開始方法を選択します。
2. アカウント設定設定を完了します。

3. ユーザーアクセスで、IAM のみのアクセスを選択します。
4. Enable を選択してプロファイルを作成します。

ユーザーは[AWS サインイン](#)を通じて AWS 変換ウェブアプリケーションにログインします。ウェブアプリケーションのユーザー識別子は、IAM プリンシパルに基づいています。

Important

セットアップの完了後に ID プロバイダーを追加することはできません。IAM Identity Center または既存の ID プロバイダーを使用してユーザーを管理するには、初期設定時にこれらのオプションのいずれかを選択します。

IAM 専用アクセスでは、プロファイルリソースに対する アクセス `transform:AccessTransformProfile` 許可を持つすべての IAM プリンシパルが AWS Transform にアクセスできます。ID プロバイダーのユーザー割り当てではなく、IAM ポリシーを使用してユーザーアクセスを管理します。

`transform:AccessTransformProfile` アクションは、AWS 変換へのアクセスのみを許可します。AWS 変換内に入ると、ワークスペースロールはユーザーが実行できるアクションを制御します。これらのロールには、管理者、コントリビューター、承認者、読み取り専用が含まれます。Workspace ロールは、ユーザーの認証方法に関係なく適用されます。

IAM 認証情報の使用

ユーザー情報の処理

IAM プリンシパルが AWS Transform にアクセスする場合、サービスは ID プロバイダーベースのアクセスとは異なる方法でユーザー ID を処理します。

ユーザー ID

IAM ユーザーは、[IAM プリンシパル](#)によって識別されます。

デュアル ID

ID プロバイダー ID (IAM Identity Center など) と IAM セッションの両方を持つユーザーは、AWS Transform で 2 人の個別のユーザーとして表示されます。これらの作業は、2 つの ID 間でリンクされません。

E メール通知

E メール通知は SSO ユーザーに対してのみ有効になります。IAM プリンシパルはワークスペースに追加できますが、E メール通知は受信されません。

監査

IAM API コールは、呼び出し元の IAM ID AWS CloudTrail でログインします。ジョブで IAM ユーザーが実行するアクションは、AWS 変換作業ログでも追跡できます。

ID プロバイダーとの IAM アクセスの有効化

IAM Identity Center またはサードパーティーの ID プロバイダーで AWS Transform を設定する場合は、認証の追加方法として IAM アクセスを有効にすることもできます。これにより、IAM プリンシパルは既存の ID プロバイダーユーザーと一緒に AWS Transform にアクセスできます。

IAM アイデンティティセンターまたはサードパーティーの ID プロバイダーでプロファイルを作成すると、IAM アクセスはデフォルトで有効になります。IAM アクセスをいつでも有効または無効にするには：

1. AWS 変換コンソールで、設定に移動します。
2. IAM 認証情報を使用したアクセス AWS 変換で、IAM アクセスの有効化をオンまたはオフに切り替えます。

IAM アクセスを有効にすると、ID プロバイダーユーザーと IAM プリンシパルの両方が AWS Transform を使用できます。IAM ユーザーと ID プロバイダーのユーザーは、同じワークスペースでコラボレーションできます。

の開始方法 AWS Organizations

AWS 変換を設定するには、次の手順に従います。

1. AWS Organizations 管理アカウントにサインインします。
2. Transform AWS サービスに移動します。
3. 組織のサービスを有効にして AWS 変換を使用するを選択します。
4. 組織メンバーアカウントに必要なアクセス許可を設定します。
5. メンバーアカウントから AWS Transform ウェブエクスペリエンスにアクセスします。

Note

AWS ソリューションで Landing Zone Accelerator (LZA) を使用してランディングゾーンを AWS Transform for Migration 機能と共に構築するには、AWS Transform アカウントと LZA のインストールが同じ AWS Organization にある必要があります。LZA デプロイと AWS 変換デプロイに別々の Organizations IDs を使用することはサポートされていません。これは、組織管理とリソースデプロイに不整合が生じる可能性があるためです。Organizations を使用して LZA インストールをセットアップする方法については、「実装ガイド」の「[Deploy a cloud foundation to support highly-regulated workloads and complex compliance requirements](#)」を参照してください。AWS

の開始方法 AWS IAM アイデンティティセンター

AWS 変換に IAM Identity Center を使用し、ユーザーとグループを追加するには、次の手順に従います。

デフォルトでは、最初に有効にしたユーザーは AWS 変換にアクセスできません。

Note

IAM Identity Center は、セットアップされているリージョンに限定されません。Transform でサポートされていないリージョンで IAM Identity Center を既にセットアップしている場合は AWS、AWS Transform に使用できます。

1. 「IAM Identity Center [のインスタンスを有効にするには](#)」の手順に従って IAM Identity Center を設定します。

外部のエンタープライズ ID プロバイダーを使用するように IAM アイデンティティセンターを設定し、そのユーザーとグループの情報を IAM アイデンティティセンターにレプリケートします。

2. AWS コンソールで、AWS 変換を選択し、開始を選択します。
3. 組織のサービスを有効にして AWS 変換を使用するを選択します。
4. 暗号化キーを選択します。デフォルトの選択は AWS マネージドキーです。カスタムキーを使用するには:
 - a. 暗号化キーで、暗号化設定をカスタマイズを選択します。

- b. KMS AWS キーを使用する を選択します。
- c. 既存のキーを選択するか、新しいキーを作成します。
- d. 送信を選択して変更を適用し、AWS 変換を有効にするを選択します。

プロファイルの表示を選択して設定を表示します。ウェブアプリケーション URL は、ユーザーが AWS Transform 統合ウェブエクスペリエンスにアクセスするために使用します。

- 5. ナビゲーションペインでユーザーを選択し、ユーザーまたはグループの割り当てを選択します。
- 6. AWS 変換の使用を承認するユーザーまたはグループの名前を検索します。検索では、ID プロバイダーから伝播されたユーザーとグループを参照します。
- 7. グループまたはユーザーを選択し、完了を選択してから、割り当てを選択します。これらのユーザーは、AWS 変換統合ウェブインターフェイスを使用する権限があります。

サードパーティーの ID プロバイダーの使用

AWS Transform は、Azure Active Directory (IdPs) との統合をサポートしています。これにより、既存の ID 管理システムをユーザー認証に使用できます。

前提条件

サードパーティー ID プロバイダーの統合を設定する前に、ID プロバイダーのユーザーに名前、E メール、ユーザー名属性が設定されていることを確認します。

保存された情報

Transform AWS with IdPs を使用すると、 は暗号化され、保護されている最小限のユーザー情報 AWS を保存します。

保存されたユーザー情報

AWS Transform は、表示名、E メールアドレス、ユーザー名 (preferred_username)、一意のユーザー識別子など、最初のログイン時に基本的なユーザープロファイル情報を保存します。この情報は、顧客の AWS 変換プロファイル設定に応じて、顧客所有の KMS キーまたはサービス所有のキーを使用して暗号化されます。データは AWS Transform の認証データベースに保存され、最初のログインセッション中にのみ収集されます。これにより、他のユーザーをワークスペースに招待するときに検索結果が入力されます。

データライフサイクル

ユーザー情報は、Transform AWS ウェブアプリに少なくとも 1 回ログインしたユーザーに対してのみ保存され、ユーザーが AWS Transform に再度ログインせずに ID プロバイダーの情報を更新すると、古くなる可能性があります。保存されたすべてのユーザー情報は、AWS 変換プロファイルが削除されると削除されます。

クライアントシークレットストレージ

セットアップ時に提供されるクライアントシークレットは、アカウント内のサービスリンクシークレット (SLS) AWS Secrets Manager を介してを使用して保存されます。

ユーザー識別子の処理

Entra

「oid」(オブジェクト識別子) クレームを一意的ユーザー識別子として使用します。これはイミュータブルで、Microsoft テナント全体のユーザーを一意的に識別します。この値は Entra コンソールで顧客に表示され、CloudTrail ログに表示されます。

Okta ワークフォース ID

トークンタイプに応じてユーザー識別に異なるクレームを使用します。ID トークンの「sub」クレームとアクセストークンの「uid」クレームです。AWS Transform は、認証中に両方のクレームに同じ値が含まれていることを検証します。この値は、Okta コンソールで顧客に表示され、CloudTrail ログに表示されます。

Azure Active Directory (Entra ID) のセットアップ

Azure Active Directory と AWS Transform の統合を設定するには:

1. Azure ポータルに移動し、Azure Active Directory を選択します。
2. 左側のナビゲーションペインで、管理 > アプリ登録を選択します。
3. + 新規登録を選択します。
4. アプリケーション名を入力し、サポートされているアカウントタイプを選択し、リダイレクト URI を空白のままにして、登録を選択します。
5. 左側のナビゲーションで、管理 > マニフェストを選択します。
6. requestedAccessTokenVersion から null に更新²し、保存を選択します。

7. Manage > Expose an API を選択し、Add a scope を選択します。
8. デフォルトの構造 を使用してアプリケーション ID URI を作成します `api://<client-id>`。
9. スコープ を追加します `transform:read_write`。
10. 証明書またはシークレットを追加を選択し、新しいクライアントシークレットを作成します。プロファイルの作成に必要なため、この値を保存します。
11. エンドポイントを選択し、OpenID Connect メタデータドキュメントを選択して、発行者 URL を検索します。メタデータの「発行者」フィールドは発行者 URL です。
12. クライアント ID、クライアントシークレット、発行者 URL を使用して、AWS 変換コンソールでプロファイルを作成します。
13. プロファイルの作成後、プラットフォームの追加を選択し、ウェブを選択し、 を入力してリダイレクト URI を追加します `<web-application-url>/login/callback`。

Okta ワークフォース ID のセットアップ

Okta ワークフォース ID と AWS Transform の統合を設定するには:

1. Okta ワークフォース ID コンソールに移動します。
2. アプリケーション > アプリケーションを選択し、アプリケーション統合の作成を選択します。
3. OIDC - OpenID Connect とウェブアプリケーションを選択し、次へを選択します。
4. アプリケーションに名前を付け、許可タイプを認可コードのままにし、リダイレクト URIs 空白のままにして、ユーザー割り当てを設定して、保存を選択します。
5. サインオンタブに移動し、発行者を動的ではなく Okta URL に設定します。
6. クライアント ID をコピーし、Security > API に移動して認可サーバーを追加して、認可サーバーの対象者として設定します。
7. 認可サーバーで、スコープタブ `transform:read_write` の下にスコープを追加します。
8. OIDC アプリケーションがこの認可サーバーを使用し、ポリシーのルールを設定できるようにするアクセスポリシーを追加します。
9. 認可サーバー設定ページで、AWS 変換でプロファイルを作成するための発行者 URL を書き留めます。
10. アプリケーション設定から発行者 URL、クライアント ID、クライアントシークレットを使用して、AWS 変換でプロファイルを作成します。
11. プロファイルの作成後、アプリケーションの全般タブにリダイレクト URL `<web-application-url>/login/callback` として を追加します。

Note

ログアウト後に AWS 変換ウェブアプリケーションにリダイレクトする場合は、Security > API でウェブアプリケーション URL を信頼できるオリジンとして設定する必要があります。

ユーザーオンボーディング

このセクションでは、AWS Transform へのアクセス権が付与されたユーザーのエクスペリエンスについて説明します。

招待の承諾

ユーザーが AWS Transform に追加されると、以下を含む招待メールが届きます。

- 挨拶と招待に関する情報
- AWS 変換ウェブアプリケーション URL
- ユーザー名
- 招待を受け入れてパスワードを設定するリンク

アカウントを設定するには:

1. ユーザーは E メールの「招待を受け入れる」リンクをクリックします。
2. 「新規ユーザーのサインアップ」ページで、パスワードを入力して確認します。
3. パスワードは、以下を含むセキュリティ要件を満たしている必要があります。
 - 8 文字以上
 - 少なくとも 1 つの大文字
 - 少なくとも 1 つの小文字
 - 少なくとも 1 つの数値
 - 少なくとも 1 つの特殊文字
4. パスワードを作成すると、アカウントが正常に作成されたことを確認するメッセージが表示されます。

AWS 変換にサインインする

Transform にサインインするには AWS :

1. 招待メールに記載されている AWS 変換ウェブアプリケーション URL に移動します。
2. ユーザー名を入力します。
3. [次へ] を選択します。
4. パスワードを入力してください。
5. [サインイン] を選択します。

ようこそエクスペリエンス

初回ログイン時に、次の内容の「ようこそ AWS の変換」ページが表示されます。

- パーソナライズされた挨拶
- 使用可能な変換機能
- ワークスペースを作成するオプション

ようこそページには、AWS 変換で利用できる変換機能に関する情報が表示されます。

- IBM z/OS の への移行をモダナイズする AWS
- VMware ワークロードを Amazon EC2 に移行する
- .NET アプリケーションを Linux 対応のクロスプラットフォーム .NET にモダナイズする
- 移行の準備状況に関するワークロードの評価

ユーザーは、ワークスペースを作成するか、チームに既存のワークスペースに追加するよう依頼することから始めることができます。

独自の Amazon S3 バケットの使用

デフォルトでは、AWS Transform はサービスマネージド Amazon S3 バケットを使用して変換アーティファクトを保存します。代わりに独自の Amazon S3 バケットを使用して、データストレージ、暗号化、アクセスポリシーをより詳細に制御できます。

Note

メインフレーム変換用の Amazon S3 バケットがある場合は、その S3 コネクタを引き続き使用でき、この機能を使用する必要はありません。

Note

Amazon S3 バケットには、操作したアップロードと変換アーティファクトのみが保存されます。内部のシステム生成アーティファクトはバケットに保存されません。さらに、AWS Transform はバケット内のアーティファクトのインデックスを作成します。インデックス化されたデータは、サービスマネージドナレッジベースに保存されます。このナレッジベースは、強化されたチャットエクスペリエンスを提供するために使用されます。

前提条件

独自の Amazon S3 バケットを設定する前に、次の要件を満たしていることを確認してください。

- バケットは、AWS 変換が有効になっているのと同じ AWS リージョンにある必要があります。
- 必要なバケットポリシーをバケットに適用する必要があります。詳細については、「[the section called “必要なバケットポリシー”](#)」を参照してください。
- AWS Transform ウェブアプリケーションを使用する場合は、バケットで CORS を設定する必要があります。詳細については、「[the section called “必要な CORS 設定”](#)」を参照してください。
- カスタム KMS AWS キーを使用する場合は、必要なキーポリシーを適用する必要があります。詳細については、「[the section called “KMS キーポリシー \(オプション\)”](#)」を参照してください。

サポートされている暗号化設定

バケットは、次のいずれかの暗号化設定を使用する必要があります。

- SSE-S3 (AES256) — KMS キーフィールドを空のままにします。AWS Transform は暗号化ヘッダーを指定せずにオブジェクトを書き込み、Amazon S3 はバケットのデフォルトの暗号化を S3-managed キーで適用します。詳細については、「Amazon S3 ユーザーガイド」の「[Amazon S3 マネージドキーによるサーバー側の暗号化 \(SSE-S3\) の使用](#)」を参照してください。

- カスタマーマネージド KMS キーを使用した SSE-KMS — KMS キー ARN を指定します。AWS Transform は、指定されたキー `x-amz-server-side-encryption: aws:kms` を使用して オブジェクトを書き込みます。 [the section called “KMS キーポリシー \(オプション\)”](#) に従います。

ただし、以下はサポートしていません。

- AWS マネージド KMS キーを使用する SSE-KMS (例: `aws/s3`)
- SSE-C
- DSSE-KMS
- 非対称 KMS キーまたは SYMMETRIC_DEFAULT 以外のキー仕様の KMS キー

 Warning

AWS マネージド KMS キーはサポートしていません。バケットが で暗号化されている場合は `aws/s3`、AWS 変換でバケットを使用する前に、カスタマーマネージドキーまたは SSE-S3 に移行します。

Amazon S3 バケットの設定

AWS Transform コンソールから独自の Amazon S3 バケットを使用するように AWS Transform を設定できます。

独自の Amazon S3 バケットを使用するには

1. AWS 変換コンソールで、設定を選択します。
2. アーティファクトストレージで、自分の S3 バケットを使用する を選択します。バケットは、現在のアカウント AWS アカウント または別のアカウントで使用できます。
3. バケットには、Amazon S3 URI を入力します。
4. (オプション) KMS キーの場合は、KMS AWS キー ARN を入力してバケット内のオブジェクトを暗号化します。このフィールドを空のままにすると、AWS Transform は暗号化ヘッダーなしでオブジェクトを書き込み、Amazon S3 はバケットのデフォルトの暗号化設定を適用します。詳細については、「Amazon S3 ユーザーガイド」の「[Amazon S3 バケット向けのサーバー側のデフォルトの暗号化動作の設定](#)」を参照してください。

バケットのデフォルトの暗号化が SSE-KMS の場合は、このフィールドにカスタマー管理の KMS キー ARN を入力し、キーポリシーを更新します。詳細については、「[the section called “KMS キーポリシー \(オプション\)”](#)」を参照してください。

5. Save. AWS Transform は、設定を適用する前にバケット設定とアクセス許可を検証します。

設定を保存すると、AWS Transform はバケットを使用して変換アーティファクトを保存します。

Note

AWS Transform は、保存時にのみ設定を検証します。バケットポリシー、CORS 設定、または KMS キーポリシーを後で変更する場合は、プロファイル設定を再保存して再検証します。

必要なバケットポリシー

Transform AWS アクセスを許可するようにバケットポリシーを設定する必要があります。Amazon S3 バケットに次のバケットポリシーを追加して、AWS 変換サービスプリンシパルが変換アーティファクトの読み取り、書き込み、削除、一覧表示できるようにします。

```
{
  "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"
            }
        }
    }
]
}

```

以下の値を置き換えます:

- *bucket-name* – Amazon S3 バケットの名前。
- *profile-account-id* – AWS 変換プロファイルに関連付けられた AWS アカウント ID。
- *profile-arn* – Transform AWS プロファイルの ARN。

必要な CORS 設定

AWS Transform ウェブアプリケーションを使用する場合は、Amazon S3 バケットで Cross-Origin Resource Sharing (CORS) を設定する必要があります。AWS Transform を有効にした後、ウェブアプリケーションドメインを見つけることができます。詳細については、「[開始方法](#)」を参照してください。

```

[
    {
        "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
}
]
```

以下の値を置き換えます:

- *webapp-domain* – ウェブアプリケーションのオリジン URL (例:
https://1a2b3c4d5e6f7a8b9.transform.us-east-1.on.aws)。これは AWS、変換設定
ページにあります。末尾にスラッシュを含めないでください。

KMS キーポリシー (オプション)

このセクションは、バケットがカスタマー管理の KMS キーで暗号化されている場合にのみ適用されます。バケットが SSE-S3 を使用している場合は、このセクションをスキップします。

バケット暗号化に独自の KMS AWS キーを指定する場合は、キーポリシーに次のステートメントを追加します。AWS Transform は、フォワードアクセスセッション (FAS) に依存してサービスプリン

シパルの許可を作成します。これらの許可は、Amazon S3 バケット内のキーアーティファクトとアクセスアーティファクトを検証するために使用されます。

```
{
  "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"
    }
  }
}
```

以下の値を置き換えます:

- *key-owner-account-id* – KMS キーを所有する AWS アカウント ID。
- *region* – AWS 変換が有効になっている AWS リージョン (例: us-east-1)。

デフォルトバケットへの切り替え

サービスマネージドバケットにはいつでも切り替えることができます。

マネージドストレージの AWS 変換に切り替えるには

1. AWS 変換コンソールで、設定を選択します。
2. アーティファクトストレージで、編集を選択します。
3. AWS マネージドストレージの変換を選択します。
4. [保存] を選択します。

⚠ Warning

変換ジョブの進行中にストレージ設定を切り替えると、それらのジョブは失敗します。進行中のジョブによって既に生成されたアーティファクトにもアクセスできません。

⚠ Important

サービスマネージドバケットに戻す場合:

- バケット内のアーティファクトは自動的に移行されません。アーティファクトを保持するには、Amazon S3 バケットからアーティファクトをダウンロードし、AWS 変換ウェブアプリケーションを介して対応するワークスペースとジョブのアーティファクトストアに再アップロードします。
- AWS Transform は、KMS キーに対する許可を廃止します。バケットに書き込まれたオブジェクトはバケットに残ります。不要になった場合は手動で削除します。

ファイルを バケットに直接アップロードする

AWS Transform ウェブアプリケーションを使用せずに、ファイルを Amazon S3 バケットに直接アップロードできます。アップロードしたファイルを変換エージェントが使用できるようにするには、ジョブの User Uploads フォルダにアップロードします。このフォルダのディレクトリパスは、次の形式を使用します。

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

以下の値を置き換えます:

- *workspace-id* – AWS 変換ワークスペースの ID。
- *job-id* – 変換ジョブの ID。

AWS Transform の各ジョブには独自の User Uploads フォルダがあります。このパスにアップロードしたファイルはウェブアプリケーションに表示され、変換エージェントがジョブを完了する間使用できます。

ファイルパスには、...、//、または先頭または末尾のスペースを含めることはできません。これらの制約に違反するファイルはエージェントに表示されません。

⚠ Important

バケットが SSE-KMS を使用している場合は、直接アップロードごとに次のヘッダーを含めます。

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

オブジェクトは、AWS 変換プロファイルに登録された KMS キーで暗号化する必要があります。他のキーで暗号化されたオブジェクトは、AWS 変換にアクセスできません。

互換性マトリックス

次の表は、AWS Transform と一般的な Amazon S3 バケット機能の互換性をまとめたものです。

機能	互換性	注意事項
バケットバージョニング	サポート	ライフサイクルルールを使用して最新でないバージョンを管理します。
オブジェクトの所有権	必須: バケット所有者の強制	バケット所有者が優先し、オブジェクトライターはサポートされていません。
ACL	サポートされていません	バケットで ACLs を無効にする必要があります。
オブジェクトロック (ガバナンスモード)	一部サポートされています	削除は保持設定によってブロックされる場合があります。
オブジェクトロック (コンプライアンスモード)	サポートされていません	—
ブロックパブリックアクセス	サポート	—
リクエスト支払い	サポートされていません	—

機能	互換性	注意事項
S3 アクセスポイント	サポートされていません	バケット名を使用します。
S3 Object Lambda アクセスポイント	サポートされていません	バケット名を使用します。
S3 on Outposts	サポートされていません	—
S3 Express One Zone (ディレクトリバケット)	サポートされていません	—
クロスリージョンバケット	サポートされていません	—
クロスアカウントバケット	サポート	—
クロスアカウント KMS キー	サポートされていません	—
ライフサイクルルール	サポート	—
CloudTrail データイベント	サポート	—

AWS 変換を有効にする

AWS 変換を有効にするには:

1. AWS マネジメントコンソールにサインインします。
2. コンソールの上部にある検索バーで、AWS 変換を検索します。
3. 検索結果からAWS 変換を選択します。

4. 開始方法 を選択して、現在のリージョンでサービスを有効にします。
5. オプション: IAM Identity Center を設定します。また、後のステップで [サードパーティー ID プロバイダー \(IdP\)](#) の使用を選択することもできます。
6. 暗号化キーを選択する: デフォルトの AWS マネージドキーまたは暗号化設定をカスタマイズします。
7. オプション: 独自の Amazon S3 バケットを使用します。デフォルトでは、AWS Transform はサービスマネージド Amazon S3 バケットを使用して変換アーティファクトを保存します。代わりに独自のバケットを使用することもできます。詳細については、「[独自の Amazon S3 バケットの使用](#)」を参照してください。
8. 有効にする AWS 変換機能を選択します。
 - [カスタム変換](#)を作成および実行するために必要なコマンドラインインターフェイス (CLI)。CLI を有効にするには、ダウンロード手順を表示して実行します。
 - ウェブアプリケーション、モダナイゼーション用のエージェントユーザーインターフェイス。ウェブアプリケーションの使用を有効にするを選択します。
9. AWS 変換を有効にするを選択します。
10. オプション: プロファイルの表示を有効にしてユーザー、設定、コネクタの AWS 変換タブにアクセスするか、ユーザーを管理してユーザーを管理するを選択します。

コンソールの左上隅にあるメニューアイコンを選択すると、ユーザー、設定、コネクタのタブにいつでもアクセスできます。
11. 次のいずれかのオプションを選択して、ユーザーアクセスを設定します。
 - IAM アイデンティティセンター
 - [サードパーティー ID プロバイダー \(IdP\)](#)
 - [IAM のみのアクセス](#)

 Note

この選択は確定され、AWS 変換を有効にすると変更できません。

12. ウェブアプリケーションの有効化 を選択します。
13. 必要なリソースが作成されている間、AWS 「変換を有効にする」と表示されます。

AWS 変換を有効にすると、設定タブに次の情報が表示されます。

- ウェブアプリケーション URL - AWS Transform ウェブアプリケーションにアクセスするための URL
- IDE の開始 URL - 統合開発環境で AWS Transform にアクセスするための URL
- リージョン - AWS 変換が有効になっている AWS リージョン

クイックスタート: AWS 変換の試行

Transform を試す最も簡単な方法は、スタンドアロン AWS アカウントを使用することです。これは proof-of-concept として、またはテスト環境で実行できます。次の手順を使用します。

1. AWS マネジメントコンソールにサインインします。
2. Transform AWS サービスに移動します。
3. 開始方法を選択してサービスを有効にします。
4. ID プロバイダーを選択してセットアップします。
5. AWS 変換サービスにユーザーを割り当てます。
6. サービスを有効にすると、AWS 変換ウェブアプリケーション URL が表示されます。
7. 新しいブラウザウィンドウでその URL を開き、AWS 変換ウェブエクスペリエンスにアクセスします。

これで、ワークスペースをセットアップする準備が整いました。

ユーザーの管理

AWS Transform へのユーザーアクセスの管理方法は、セットアップ時に選択したアクセスモデルによって異なります。IAM アイデンティティセンターを設定した場合は、IAM アイデンティティセンターを通じてユーザーを追加します。サードパーティー ID プロバイダーを設定した場合は、そのプロバイダーのユーザーを管理します。IAM のみのアクセスを選択した場合、ユーザーアクセスは IAM ポリシーによって管理されます。

IAM アイデンティティセンターへのユーザーの追加

IAM Identity Center でユーザーを追加するには:

1. IAM Identity Center コンソールに移動します。

2. ナビゲーションペインで [Users] (ユーザー) を選択します。
3. [Add user] (ユーザーを追加) を選択します。
4. 必要な情報を入力します。
 - ユーザー名 - ユーザーの一意的識別子 (後で変更することはできません)
 - E メールアドレス - ユーザーの E メールアドレス
 - 名と姓 - ユーザーの名前
 - 表示名 - ユーザーリストに表示される名前
5. パスワード では、ユーザーがパスワードを受け取る方法を選択します。
 - E メールを送信する - セットアップ手順を E メールで送信する
 - ワンタイムパスワードの生成 - 手動で共有するパスワードを作成する
6. 次へ を選択して、ユーザー情報を確認します。
7. 詳細を確認し、ユーザーの追加を選択します。

ユーザーを追加すると、IAM Identity Center アカウントを設定するための招待メールが届きます。招待リンクは 7 日間有効です。

この動画では、IAM Identity Center と AWS Transform の使用についても説明します。

[VideoTitle](#)

AWS 変換へのユーザーの追加

IAM Identity Center にユーザーを追加したら、AWS Transform へのアクセス権を付与できます。

1. AWS 変換コンソールに戻ります。
2. ナビゲーションペインで、ユーザーとグループを選択します。
3. ユーザータブまたはグループタブを選択します。
4. IAM Identity Center から追加するユーザーまたはグループを検索して選択します。
5. ユーザーとグループの割り当て を選択して、選択したユーザーまたはグループに AWS 変換へのアクセス権を付与します。

ユーザーを追加すると、招待を受け入れてサインインするまで、ステータスが「保留中」のユーザーリストに表示されます。

IAM のみのアクセスを持つユーザーの管理

IAM のみのアクセスで AWS 変換を設定した場合、ユーザーアクセスは IAM ポリシーによって管理されます。プロファイルリソースに対する `transform:AccessTransformProfile` 許可を持つすべての IAM プリンシパルは、AWS Transform にアクセスできます。

ユーザーまたはロールに AWS Transform へのアクセスを許可するには:

1. [IAM console] (IAM コンソール) に入ります。
2. `transform:AccessTransformProfile` アクションを含むポリシーを IAM ユーザーまたはロールにアタッチします。ポリシーの例については、[「IAM 認証情報を使用した AWS 変換へのアクセスをユーザーに許可する」](#)を参照してください。

アクセスを取り消すには、IAM ユーザーまたはロールからポリシーを削除します。

共同作業者のアクセス許可について

AWS Transform はワークスペースベースのアクセス許可モデルを使用して、リソースとアクションへのアクセスを制御します。各ユーザーにはワークスペース内の特定のロールが割り当てられ、実行できるアクションが決まります。ユーザーは、異なるワークスペースで異なるロールを持つことができます。

ユーザーロール

AWS Transform は、各ワークスペース内で 5 つのユーザーロールをサポートします。これらのロールはワークスペースのコンテキスト内に適用され、ユーザーはメンバーである各ワークスペースにロールが割り当てられます。各ロールに定義されているアクセス許可はワークスペースに依存しないため、ワークスペース A の管理者ロールを持つユーザー A には、ワークスペース B の管理者ロールを持つユーザー B と同じアクセス許可があります。

ロールのアクセス許可

各ロールの詳細なアクセス許可:

アクション	ResourceType	管理者	承認者	コントロ ビュー ター	ReadOnly
作成	ワークスペース	✓	✓	✓	✓

アクション	ResourceType	管理者	承認者	コントローラー	ReadOnly
リスト	ワークスペース	✓	✓	✓	✓
Get	ワークスペース	✓	✓	✓	✓
更新	ワークスペース	✓	✗	✗	✗
Delete	ワークスペース	✓	✗	✗	✗
作成	ChatMessage	✓	✓	✓	✗
読み取り	ChatMessage	✓	✓	✓	✓
作成	RoleAssociation	✓	✗	✗	✗
読み取り	RoleAssociation	✓	✓	✓	✓
更新	RoleAssociation	✓	✗	✗	✗
Delete	RoleAssociation	✓	✗	✗	✗
読み取り	CriticalHITLTask	✓	✓	✓	✓
更新	CriticalHITLTask	✓	✓	✗	✗
Delete	CriticalHITLTask	✓	✓	✗	✗
読み取り	HITLTask	✓	✓	✓	✓
更新	HITLTask	✓	✓	✓	✗
Delete	HITLTask	✓	✓	✓	✗
作成	ジョブ	✓	✓	✓	✗
読み取り	ジョブ	✓	✓	✓	✓
更新	ジョブ	✓	✓	✓	✗

アクション	ResourceType	管理者	承認者	コントロ ビュー ター	ReadOnly
Delete	ジョブ	✓	✓	✓	✗
読み取り	作業ログ	✓	✓	✓	✓
作成	アーティファクト	✓	✓	✓	✗
読み取り	アーティファクト	✓	✓	✓	✓
更新	アーティファクト	✓	✓	✓	✗
Delete	アーティファクト	✓	✓	✓	✗
作成	コネクタ	✓	✓	✓	✗
読み取り	コネクタ	✓	✓	✓	✓
更新	コネクタ	✓	✓	✓	✗
Delete	コネクタ	✓	✓	✓	✗

ヒューHuman-in-the-loop (HITL) アクション

AWS Transform には、標準アクションとクリティカルアクションの 2 種類の HITL アクションがあります。

標準 HITL アクション

これらは、コントリビューター、承認者、または管理者ロールを持つユーザーが実行できる日常的なアクションです。

重要な HITL アクション

これらは大きな影響を与えるアクションであるため、より高いアクセス許可レベルが必要です。以下に例を示します。

- メインブランチへのコードのマージ
- グラフ分解の実行

- 本番環境へのコードのデプロイ

重要な HITL アクションは、承認者または管理者ロールを持つユーザーのみが実行できます。

AuthZ ポリシーで標準 HITL アクションとクリティカル HITL アクションを区別するために、AWS Transform には 2 つの個別の HITL APIs が用意されています。1 つは標準 HITL アクションを完了するための API、もう 1 つはクリティカル HITL アクションを完了するための API です。

AWS 変換環境

AWS 変換は、自然言語処理を使用してワークロード変換を計画および実行するのに役立つエージェントサービスです。主に会話インターフェイスを介して AWS 変換を操作します。このインターフェイスでは、サービスはディスカッションのコンテキストに基づいて適応し、応答します。たとえば、「オンプレミス VMs」などの日常的な言語を使用して、現在のアーキテクチャとトランスフォーメーションの目標を説明することから始めることができます。EC2

AWS Transform と会話すると、サービスは特定の要件に合ったカスタマイズされたジョブプランを構築します。会話フローは動的で、ユーザーによって駆動されるため、変換計画を反復的に調整できます。会話中にジョブプランを変更するには、「本番環境のカットオーバー前にテストフェーズを追加する」や「ソリューションがすでにあるためバックアップステップを削除する」などのリクエストを行います。また、前のタスクに戻って再度実行することもできます。AWS Transform は、入力に基づいて計画を継続的に更新し、最終的な変換戦略が技術的およびビジネス上のニーズを満たすようにします。

Transform を開くと、次のようになります AWS。

- ビューコントロールペイン: これは AWS 変換の左側にある狭いペインです。アイコンのいずれかを選択して、ビューコントロールペインの右側に表示されるビューを選択できます。各アイコンにカーソルを合わせると、ビューを説明するツールヒントが表示されます。give 標準ビューは、上から下まで次のように表示されます。
 - ジョブプラン
 - ダッシュボード
 - 承認
 - アーティファクト
 - 作業ログ

一部のワークフローでは、追加のビューが提供されます。

ジョブプランで作業している場合、ビューは次のとおりです。

- Chat
- 承認
- アーティファクトストア
- 作業ログ

ワークスペースでは、ビューが上部に表示されます。

- ジョブ
 - アーティファクト
 - コラボレーター (ユーザー)
 - Connector
 - 設定
- ビューペインはビューコントロールの横にあります。
- チャットペインは中央にあります。ここで、AWS Transform との会話を行います。

Note

読み取り専用アクセス許可を持つユーザーは、チャットでメッセージを送信できません。

- 右側にはコラボレーションペインがあります。これは、次のようなヒューマンインザループ (HITL) アクティビティが実行されたときに表示されます。
 - データファイルのアップロード
 - AWS Transform が提供する情報と計画の確認

言語設定

AWS Transform は、一部のワークフローのローカリゼーションをサポートしています。右上隅の設定アイコンを選択し、言語メニューから任意の言語を選択することで、ウェブアプリケーションの表示言語を変更できます。

次の言語がサポートされています。

- 英語 (米国)
- 英語 (英国)

- ドイツ語 (ドイツ語)
- スペイン語 (スペイン語)
- フランス語 (フランス語)
- 日本語 (日本語)
- バハサインドネシア (インドネシア)
- イタリア語 (イタリア語)
- ポルトガル語 (ポルトガル語)
- 「"❖❖"❖❖ (韓国語)
- 中文 (簡体) (簡体字中国語)
- 中文 (繁体) (繁体字中国語)
- トルコ語 (トルコ語)

次のワークフローはローカリゼーションをサポートしています。

- [移行 \(VMware を含む\)](#)

Note

その他の AWS 変換ワークフローは現在、英語でのみ利用できます。追加のワークフローのローカリゼーションサポートは、今後の更新で追加される可能性があります。

プロジェクトを開始する

AWS 変換は、モダナイゼーションプロジェクトと移行プロジェクトをガイドします。開始するには、以下の手順を実行します。

1. プロジェクトをホストするワークスペースを作成します。[ワークスペース] ページで、[ワークスペースを作成] をクリックします。手順に従ってワークスペースを開きます。
2. チャットペインに「ジョブの作成」と入力します。
3. ジョブを選択し、AWS Transform の指示に従います。入力が必要な場合は、コラボレーションペインが表示され、必要なものが説明されます。

ワークフローの柔軟性

AWS Transform ワークフロー環境は、モダナイゼーションプロジェクトの進行に柔軟性を提供します。チャットペインで自然言語を使用すると、次のことができます。

- Transform AWS がすでに完了しているタスクを再試行し、途中で異なるヒューman-in-the-loop 入力を提供します。
- モダナイゼーションソースに変更があった場合などは、ジョブを再実行します。

AWS 変換とのチャット

AWS 変換チャットは、プロジェクトのすべての段階で利用できます。チャットを開くには、ウェブコンソールの右下隅にある紫色の六角形アイコンをクリックします。

チャットには、何でも質問できます。たとえば、チャットに概念の説明、プロセスのガイド、AWS 変換リクエストまたはレスポンスの説明、AWS 変換レポートの説明を求めることができます。

Note

読み取り専用アクセス許可を持つユーザーは、チャットでメッセージを送信できません。

AWS チャット統合を変換する

AWS 変換チャットは以下と統合されています。

エクスペリエンスベースのアクセラレーション

[エクスペリエンスベースのアクセラレーション \(EBA\)](#) は、AWS Transform チャットを通じて提供されます。これにより、Windows ワークロードの EBA 評価を実行し、プランを生成できます。まず、CAST から評価結果をインポートします。これにより、アプリケーションポートフォリオを検出し、AWS Transform Chat を使用してビジネスニーズを満たすアプリケーションを選択できます (たとえば、複雑さやコード行に基づいてアプリケーションをフィルタリングするなど)。その後、選択したアプリケーションのより詳細な評価を実行し、モダナイゼーション計画を生成できます。

AWS カウントダウンプレミアム

AWS Transform に統合された[AWS カウントダウンプレミアム \(CDP\)](#) は、指定された AWS エンジニアリングサポートにより、持続的なエキスパートガイダンスを提供します。指定された CDP エン

エンジニアが技術スタックを深く掘り下げ、問題が発生した場合にパーソナライズされたコンテキスト対応サポートを提供します。ユーザーは AWS Transform を活用して、チャットインターフェイスから直接サポートチケットを記録できます。サポートチケットには、作業ログとジョブプランの詳細が埋め込まれ、サポートが顧客のコンテキストを理解するのに役立ちます。顧客またはパートナーが CDP プログラムに参加している場合、サポートチケットは指定された CDP エンジニアに自動的にルーティングされ、問題のデバッグ、変換出力の解釈、迅速な問題解決の進行状況の促進に役立ちます。例えば、次のようになります。

- .NET ワークロードの場合、CDP はリポジトリコネクタの問題、依存関係の解決、および変換後のデプロイを支援できます。
- メインフレームシナリオでは、CDP はリファクタリングされた Java コードのトラブルシューティング、データベース移行ツールの設定、CI/CD パイプラインの構築に役立ちます。
- VMware 移行では、CDP はネットワーク設定、MGN セットアップ、エージェントのインストールを高速化できます。

AWS スキルビルダー

[AWS スキルビルダー](#)は、チャットを通じて関連する学習モジュールを提供します。学習ニーズについてトランス AWS フォームチャットに尋ねると、関連するコースカタログが表示されます。スキルビルダーは、変換ステージを進むにつれて、コンテキストに応じたマイクロラーニングエクスペリエンスを提供します。

デベロッパーツール

エージェント IDEs、エージェントプラグイン、および MCP サーバーから AWS Transform エージェントにアクセスできます。ワークフローに適した表面を選択します。

Note

Kiro Power プラグインとエージェントプラグインは MCP サーバーを自動的にインストールします。MCP サーバーを手動でインストールする必要があるのは、Kiro Power またはエージェントプラグインなしで使用する場合のみです。

Kiro Power

[AWS Transform Kiro Power](#) は、Kiro IDE から AWS Transform エージェントへのフルアクセスを提供します。

AWS Transform Kiro Power をインストールするには

1. Kiro IDE を開き、Powers パネルに移動します。
2. リストでAWS 変換を見つけてインストールします。
3. Kiro Chat を開き、電源をクリックし、電源を試すを選択します。

または、Powers パネルで Add Custom Power > Import power from GitHub を選択して貼り付けます。 <https://github.com/kirodotdev/powers/tree/main/aws-transform>

エージェントプラグイン

AWS Transform エージェントプラグインは、 `plugins/aws-transform` ディレクトリの GitHub の [awslabs/agent-plugins](https://github.com/aws-labs/agent-plugins) で利用できます。プラグインは Claude Code、Codex、および Cursor をサポートしています。

プラグインには、`.claude-plugin`、`.codex-plugin`、および `.mcp.json` の設定が含まれています。`.NET`、メインフレーム、VMware、SQL Server、カスタム変換など、すべての AWS 変換エージェントへのアクセスを提供します。

Claude Code で をインストールするには

以下の コマンドを実行します。

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

Codex またはカーソルに をインストールするには

エージェントプラグインリポジトリ README の指示に従います。

IDE プラグイン

AWS Transform IDE プラグインは、VS Code および Open VSX 互換エディタで使用できます。AWS 変換定義の作成や実行など、変換カスタム機能を公開します。また、モダナイゼーション要件、ビジネスルール、ソースコード間のトレーサビリティ、技術ドキュメントをオンデマンドで生成する機能など、メインフレームの機能についても説明します。開始するには、次のいずれかの方法を使用して をインストールします。

- VS Code: [VS Code Marketplace](#) から をインストールします。
- Open VSX: [Open VSX レジストリ](#) からインストールします。

MCP サーバー

MCP サーバーを使用して、AWS 変換を MCP 互換クライアントとプログラムで統合します。PyPI から MCP サーバーを としてインストールできます `awslabs.aws-transform-mcp-server`。ワークスペース管理、ジョブ管理、ヒューman-in-the-loop (HITL) タスク、アーティファクト処理、コネクタ管理、チャット、リソースブラウジングのための 19 のツールを提供します。

前提条件

- Python 3.10 以降

インストール

Claude コード

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


Kiro

を に追加します ~/.kiro/settings/mcp.json。

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

特定の MCP クライアントの設定の詳細については、GitHub の [aws-transform-mcp-server](#) README を参照してください。

認証

AWS Transform は、SSO (AWS Identity and Access Management アイデンティティセンター) と IAM ロールベースの認証をサポートしています。

SSO (IAM アイデンティティセンター)

AI アシスタントに「Configure AWS Transform with SSO」を依頼します。アシスタントは開始 URL を提供し、ログイン用にブラウザを開きます。

IAM ロール

環境は AWS 認証情報を自動的に提供します。MCP クライアント設定 env ブロック AWS_PROFILE でを設定して、特定のプロファイルを選択します。

AWS Transform の認証の設定については、「」を参照してください [AWS 変換の開始方法](#)。

AWS 変換用のカスタムエージェントを構築する

[エージェントビルダーツールキット](#)を使用すると、AWS パートナーと顧客は、AWS Transform 内で機能する特定のモダナイゼーションのニーズに合わせたエージェントを構築できます。

エージェントビルダーツールキットをインストールするには

1. Kiro IDE を開き、Powers パネルに移動します。

2. リストで AWS Transform Agent Toolkit を見つけてインストールします。

または、Powers パネルで Add Custom Power > Import power from GitHub を選択して貼り付けます。 <https://github.com/kirodotdev/powers/tree/main/aws-transform-agent-toolkit>

エージェントビルダーツールキットを使用すると、移行およびモダナイゼーションのコンピテンシーパートナー、ISVs、またはお客様は、特定のユースケースに対応する変換ソリューションを作成できます。特殊なエージェント、ツール、ナレッジベース、ワークフローを AWS Transform エージェント AI 機能に統合できます。

エージェントライフサイクル

構築

Kiro Power を使用して、ベースエージェントと SDK を組み合わせて独自のカスタマイズされたエージェントを記述します。

共有

チームまたはパートナーネットワーク間でエージェントを分散します。

登録

エージェントを AWS Transform に登録して、他のユーザーが検出できるようにします。

移行評価

AWS 変換評価は、オンプレミスインフラストラクチャを に移行する際のコスト、実現可能性、ビジネス価値を評価するのに役立ちます AWS。評価では、適切なサイジングに関する自動レコメンデーション、マルチシナリオ比較、チャットによるインタラクティブな絞り込みを行うことができます。

移行評価を使用して以下を行うことができます。

- Amazon RDS for SQL Server、Amazon EC2、Amazon EBS、Amazon S3、Amazon FSx のコスト見積もりを取得する
- 適切なサイジングの自動レコメンデーションを受け取る
- ビジネス価値と持続可能性への影響を評価する
- 複数の移行シナリオを並べて比較する
- チャットを通じて評価をインタラクティブに絞り込む
- エグゼクティブプレゼンテーションと詳細レポートを生成する

前提条件

移行評価を作成する前に、以下があることを確認してください。

- AWS 変換ワークスペース
- サポートされている形式でデータをインベントリするか、チャットで大まかな推定を使用できます。
- 評価を作成および実行するための適切なアクセス許可

移行評価の仕組み

移行評価は、最初のデータアップロードから最終的な成果物生成までをガイドする構造化ワークフローに従います。

移行評価を完了するには

1. Transform AWS ワークスペースで移行評価ジョブを作成します。
2. オンプレミスのインベントリデータをアップロードします。

3. AWS Transform がデータから生成する検出結果を確認します。
4. サーバーを除外するか、グループ化を調整して、インベントリの範囲を管理します。
5. 料金とインフラストラクチャの前提条件を使用して評価シナリオを設定します。
6. 評価を実行します。
7. チャットを通じて結果を確認して絞り込みます。

結果を確認したら、追加のシナリオを作成し、シナリオを比較して、プレゼンテーションやレポートなどの成果物を生成できます。

インベントリデータのアップロード

AWS Transform は、複数のソースからのインベントリデータを受け入れます。次の表に、サポートされているデータ形式を示します。

データソース	説明
AWS 検出ツールのエクスポートを変換する	AWS 変換検出ツールによって検出された自動サーバーインベントリ
RVTools	VMware 環境から ZIP/CSV または Excel 形式でエクスポートします。フルエクスポートと vInfo のみのエクスポートの両方がサポートされています。
CMDB データ	設定管理データベースのエクスポート
パートナー検出ツール	AWS パートナー検出ツールからのデータ
「Migration Evaluator」	Migration Evaluator コンソールからの Quick Insights ファイル
MPA 形式	移行ポートフォリオ評価のインポートファイル
AWS データテンプレートを変換する	AWS Transform Assessment Data テンプレートから作成された Microsoft Excel ファイル

AWS 変換は、取り込み中にファイル形式を自動的に識別します。取り込みプロセスでは、データを検証し、一部のフィールドが欠落している場合に部分的なデータを受け入れ、増分アップロードをサポートします。インベントリを補完するために、いつでも追加のファイルをアップロードできます。

検出結果の確認

AWS Transform がインベントリデータを処理すると、検出の概要が生成されます。概要には、次の情報が含まれます。

- オペレーティングシステム別のサーバー数
- 物理サーバーと仮想サーバーの内訳
- SQL Server 検出
- ストレージの概要
- データ品質に関する警告

検出結果を確認して、評価に進む前に AWS Transform がインフラストラクチャを正しく識別したことを確認します。

インベントリスコープの管理

検出結果を確認したら、評価の範囲を絞り込むことができます。インベントリの範囲を管理するには、次のアクションを使用します。

- 特定のサーバーまたはワークロードのインベントリをクエリする
- 評価スコープからサーバーを除外する
- サーバーのグループ化とアプリケーションの依存関係を確認する
- 特殊な評価のための SQL Server ワークロードを特定する

評価シナリオの設定

評価を実行する前に、AWS Transform がレコメンデーションの生成に使用する前提条件を設定します。これらの前提条件はいつでも変更でき、さまざまな設定で新しいシナリオを実行できます。

次の表に、使用可能な前提条件カテゴリを示します。

仮定	説明
料金モデル	Database Savings Plans (Amazon RDS for SQL Server の場合)、オンデマンド、またはリザーブドインスタンス

仮定	説明
ターゲット AWS リージョン	移行されたワークロードをホストする予定の AWS リージョン
インスタンスタイプの除外	レコメンデーションから除外する Amazon EC2 インスタンスファミリーまたはタイプ
Amazon EBS 設定	ストレージのボリュームタイプとパフォーマンス設定
SQL Server のライセンス	Amazon RDS for SQL Server の Bring Your Own Media (BYOM) またはライセンス込み (LI)、EC2 の SQL Server のライセンス込み (LI) または Bring Your Own License (BYOL)

評価機能

AWS 変換評価は、移行の複数のディメンションを対象としています。以下のセクションでは、各評価機能について説明します。

Amazon EC2 の適切なサイズ設定

AWS Transform はオンプレミスサーバーの仕様を分析し、適切なサイズの Amazon EC2 インスタンスを推奨します。レコメンデーションは、CPU、メモリ、パフォーマンスの要件を考慮します。

Amazon EBS ストレージ

AWS Transform は、現在のストレージの使用状況とパフォーマンス要件に基づいて Amazon EBS ボリュームタイプと設定を推奨します。

Amazon FSx

AWS Transform はファイルストレージワークロードを評価し、サポートされている Amazon FSx ファイルシステムタイプのコスト見積もりなど、Amazon FSx 移行に関する推奨事項を提供します。

Amazon RDS for SQL Server

AWS Transform は、オンプレミスの SQL Server データベースを Amazon RDS for SQL Server に移行するコストを評価します。AI を活用したエージェントを使用して、AWS Transform はオンプレミスの SQL Server 環境を分析し、コンピューティングとメモリのレコメンデーションがワークロード

の要件に一致する完全な移行ビジネスケースを数分で提供します。これにより、過剰なプロビジョニングを回避し、必要な分だけ料金を支払うことができます。

RDS for SQL Server の評価には、次の機能が含まれています。

- 既存の SQL Server ライセンスを使用できるようにする Bring Your Own Media (BYOM) ライセンス
- ライセンス込み (LI) ライセンスオプション
- Database Savings Plans を使用したコスト最適化。オンデマンド料金と比較して最大 20% のコスト削減を実現
- AWS 移行コストを相殺するためのクレジットとサポートを提供する移行促進プログラム (MAP) の適格性評価

RDS for SQL Server の評価は、サポートされている任意のデータ形式で開始できます。これには、RVTools のエクスポート、設定管理データベース (CMDB) データ、AWS 変換検出ツールからのエクスポート、その他のサードパーティーの検出ツールが含まれます。What-If シナリオを作成して、複数のコストモデルをリージョン、リソース使用率、料金条件などのカスタマイズされた前提と比較します。

プロンプトの例:

- 「SQL Server データベースを RDS for SQL Server に移行するコストを見積もる」
- 「RDS for SQL Server の BYOM とライセンス込み料金の比較」
- 「RDS ワークロードの Database Savings Plans オプションを表示する」

Amazon EC2 SQL Server

AWS Transform は SQL Server ワークロードを評価し、ライセンス分析や専用ホストマッピングなど、Amazon EC2 で SQL Server を実行するための推奨事項を提供します。

オンプレミス料金

AWS Transform は、現在のオンプレミスコストを見積もり、AWS 料金と比較するためのベースラインを提供します。チャットを通じて全体的なコストを調整できます。オンプレミスのコスト調整は PDF レポートとチャットレスポンスに反映されますが、PPTX エクスポートには反映されません。

プロンプトの例:

- 「オンプレミスサーバーのコストをサーバーあたり 1 か月あたり 500 USD に更新する」
- 「データセンターの年間施設コスト 50,000 USD を追加」

持続可能性

AWS Transform は、への移行によるカーボンフットプリントの削減を推定します AWS。チャットを通じて持続可能性メトリクスを調べることができます。

次のようなプロンプトを試してください。

- 「この移行の推定炭素削減量を教えてください」
- 「のワークロードのエネルギー効率はどう向上しますか AWS?」

ビジネス価値評価

AWS Transform は、インフラストラクチャのコスト削減を超えた移行のより広範なビジネス価値を評価します。この評価では、スタッフの生産性、レジリエンス、ビジネスの俊敏性について説明します。

プロンプトの例を次に示します。

- 「への移行によるスタッフの生産性の向上を見積もる AWS」
- 「移行後にどのような耐障害性の向上が期待できますか？」
- 「この移行によるビジネス俊敏性のメリットを教えてください」

ネットワークコスト

AWS Transform は、データ転送や接続要件など、移行に伴うネットワーク関連のコストを見積もります。

サポートコスト

AWS Transform には、選択した AWS サポート階層に基づいて評価にサポートプランのコストが含まれます。

エンドユーザーコンピューティング

AWS Transform は、エンドユーザーのコンピューティングワークロードを評価し、AWS エンドユーザーのコンピューティングサービスにレコメンデーションを提供します。

チャットベースの評価の使用

チャットを通じて AWS Transform とやり取りして、評価を作成および絞り込むことができます。チャットは 3 つのインタラクションモードをサポートしています。

会話を続けることで、評価を反復的に絞り込むことができます。AWS Transform はメッセージ間のコンテキストを維持し、入力に基づいて評価結果を更新します。

限られたデータによる大まかな推定

詳細なインベントリファイルをアップロードせずにチャットで環境を記述することで、概算コストの見積もりを取得できます。概算見積りはスタンドアロン機能であり、アップロードされたインベントリデータやチャットによるデータエンリッチメントと組み合わせることはできません。

次のようなプロンプトを使用できます。

- 「Windows サーバー 200 台、Linux サーバー 150 台、推定コスト AWS」
- 「500 VMs AWS」
- 「平均 8 CPUs」

チャットによるインベントリの追加

チャットから直接インベントリの詳細を指定して、アップロードされたファイルを補足または置き換えることができます。

次のようなプロンプトを使用します。

- 「128 GB RAM の 4 つの CPUs」
- 「4 つの CPUs」

コストの変更とサービスの追加

チャットを通じて、オンプレミスのコストを調整し、評価に AWS サービスを追加できます。

オンプレミス調整のプロンプト例:

- 「今後のハードウェア更新を考慮してオンプレミスコストを 15% 増やす」
- 「オンプレミスベースラインに年間 100,000 USD のメンテナンスコストを追加する」

Transform AWS 評価で完全にサポートされていない AWS サービスの概算コスト見積りを追加することもできます。これにより、より完全な分析が得られますが、これらの見積もりは自動レコメンデーションよりも精度が低くなります。

- 「移行されたすべてのサーバーの AWS バックアップコストを追加する」
- Amazon CloudWatch のモニタリングコストを見積りに含める」
- 「10 Gbps 接続の AWS Direct Connect コストを追加する」

シナリオの比較

さまざまな前提で複数の評価シナリオを作成し、それらを比較して最適な移行戦略を特定できます。

シナリオの作成

次のようなプロンプトを試してください。

- 「SQL Server データベースを RDS for SQL Server に移行するための BYOM シナリオを作成する」
- 「RDS for SQL Server の Database Savings Plans を使用してシナリオを作成する」
- EC2 上のすべての SQL Server ワークロードの LI シナリオを作成する」
- 「us-west-2 のすべてのワークロードでシナリオを作成する」

比較の実行

プロンプトの例を次に示します。

- 「すべてのシナリオでコスト比較を表示する」
- 「総所有コストが最も低いシナリオはどれですか？」

What-if 分析

まったく新しいシナリオを作成せずに、特定の変更の影響をモデル化します。

プロンプトの例:

- 「SQL Server ワークロードを EC2 ではなく RDS for SQL Server に移動するとどうなりますか？」
- 「BYOM と RDS for SQL Server のライセンス込みのコスト差は何ですか？」
- 「最小 50 台のサーバーを移行から除外するとどうなりますか？」
- 「ストレージ最適化インスタンスのみを使用する場合はどうなりますか？」
- 「gp3 ボリュームから io2 ボリュームに移行するとどのような影響がありますか？」

成果物の生成

評価が完了したら、複数の形式で成果物を生成できます。次の表に、使用可能な出力形式を示します。

形式	説明	カスタマイズ
PPTX (PowerPoint)	結果の概要と推奨事項を含むエグゼクティブプレゼンテーション	固定構造
XLSX (Excel)	サーバーレベルのレコメンデーションとコスト内訳を含む詳細なデータエクスポート	固定構造
PDF	レポートドキュメント	チャットでカスタマイズ可能

関連トピック

- [開始方法](#)
- [カスタムジョブ](#)
- [検出ツール](#)

カスタム

AWS Transform custom とは

AWS Transform Custom は、エージェント AI を使用してソフトウェア、コード、ライブラリ、フレームワークの大規模なモダナイゼーションを実行し、技術的負債を削減します。言語バージョンのアップグレード、API とサービスの移行、フレームワークのアップグレードと移行、コードリファクタリング、組織固有の変換など、さまざまなシナリオを処理します。

継続的な学習を通じて、エージェントはすべての実行とデベロッパーのフィードバックから改善し、専門的な自動化の専門知識を必要とせずに、高品質で反復可能な変換を提供します。

主な機能

AWS 変換カスタムには次の機能があります。

- 自然言語駆動型変換定義 - 自然言語、ドキュメント、コードサンプルを使用してカスタム変換を作成する
- 変換の実行 - 複数のコードベースに一貫して確実に変換を適用する
- 継続的な学習 - すべての実行から変換品質を自動的に改善する
- AWSマネージド変換 - 一般的なシナリオでready-to-use AWS、検証済みの変換を使用します。

変換パターン

AWS 変換カスタムは、モダナイゼーションのニーズに対応するための多様な変換パターンをサポートします。各パターンには、必要な変更の範囲と性質に基づいて異なる複雑さの特性があります。

パターン	説明	複雑さ	例
API とサービスの移行	機能を維持しながら API バージョンまたは同等のサービスを移行する	中	AWS SDK v1→v2 (Java、Python、JavaScript)、Boto2→Boto3、JUnit 4→5、javax→jakarta

パターン	説明	複雑さ	例
言語バージョンのアップグレード	同じプログラミング言語の新しいバージョンへのアップグレード、新機能の採用、廃止された機能の置き換え	低～中	Java 8 → 17、Python 3.9 → 3.13、Node.js 12 → 22、TypeScript バージョンアップグレード
フレームワークのアップグレード	同じフレームワークの新しいバージョンにアップグレードし、重大な変更に対処する	中	Spring Boot 2.x→3.x、React 17→18、Angular アップグレード、Django アップグレード
フレームワークの移行	同様の目的を果たすまったく異なるフレームワークへの移行	高	Angular →React、Redux →Zustand、Vue.js →React
ライブラリと依存関係のアップグレード	同じ言語とフレームワークを維持しながら、サードパーティーライブラリを新しいバージョンにアップグレードする	低～中	Pandas 1.x→2.x、NumPy アップグレード、Hadoop/HBase/Hive ライブラリアップグレード、Lodash アップグレード
コードリファクタリングとパターンモダナイゼーション	外部機能を変更せずにコードパターンをモダナイズし、ベストプラクティスを採用する	低～中	Print→Logging フレームワーク、文字列連結→f-strings、型ヒントの採用、オブザーバビリティ計測

パターン	説明	複雑さ	例
スクリプトおよびFile-by-File変換	ファイルの大部分が自己完結型である独立したスクリプトまたは設定ファイルの翻訳	低～中	AWS CDK → Terraform、Terraform → CloudFormation、Excel → Python ノートブック、Bash → PowerShell
アーキテクチャの移行	最小限のコード変更でハードウェアアーキテクチャまたはランタイム環境を移行する	やや高い	x86 → AWS Graviton (ARM)、オンプレミス → Lambda、従来のサーバー → コンテナ
Language-to-Language移行	あるプログラミング言語から別のプログラミング言語へのコードベースの翻訳	非常に高い	Java → Python、JavaScript → TypeScript、C → Rust、Python → Go
カスタムおよび組織固有の変換	独自の組織要件と特殊なモダナイゼーションのニーズ	可変	カスタム内部ライブラリの移行、組織固有のコーディング標準、独自のフレームワークの移行

Note

COBOL/メインフレーム言語の場合は、メインフレームの AWS 変換を使用します。.NET Framework を .NET Core にアップグレードするには、Windows の AWS 変換を検討してください。VMware への移行については AWS、AWS Transform for VMware を検討してください。

AWS Transform カスタムの仕組み

AWS 変換カスタムは通常、複数のコードベースまたはモジュールが変換される大規模なプロジェクトで使用されます。チームは通常、次の 4 フェーズのワークフローに従います。

変換の定義 - エージェントに自然言語プロンプト、ドキュメント、コードサンプルを提供し、最初の変換定義を生成します。この定義は、チャットまたは直接編集によって繰り返し改良できます。AWS マネージド変換を使用する場合、このフェーズはスキップできます。

パイロットまたは Proof-of-Concept - サンプルコードベースで変換をテストし、結果に基づいて絞り込みます。この検証フェーズは、完全な変換のコストと労力を見積もるのに役立ちます。このフェーズでは、継続的な学習により品質が向上します。

スケーリングされた実行 - CLI を使用して自動一括実行を設定し、開発者が結果を確認して検証します。ウェブアプリケーションを使用して進行状況をモニタリングし、複数のリポジトリ間の変換を追跡します。

モニタリングとレビュー - 継続的な学習により、変換の品質が自動的に向上します。以前の実行から抽出された教訓を確認して品質基準を満たしていることを確認し、役に立たないものをアーカイブします。

の重要な概念の理解

このセクションでは、AWS 変換カスタムを使用するための主要な概念について説明します。

変換の定義

変換定義には、特定のコード変換を実行するために必要な手順と知識が含まれています。スキルとして表され、以下で構成されます。

- **SKILL.md (必須)** - コア変換ロジックと実行手順を含む YAML フロントマター (name および description フィールド) を使用した Markdown の手順
- **references/ folder (オプション)** - 変換の実行中に必要に応じてロードされたドキュメント
- **scripts/ folder (オプション)** - 実行中に変換がダウンロードして実行するスクリプト

AWS Transform CLI は、実行、検査、または変更が必要なときに、変換定義を現在のディレクトリに自動的にダウンロードします。

⚠ Important

変換を公開する場合、ディレクトリには SKILL.md のみを含める必要があり、オプションで references/フォルダ、scripts/フォルダ、またはその両方を含める必要があります。他のファイルやサブディレクトリは許可されません。

ℹ Note

CLI バージョン 2.0 以降では、スキル形式が使用されます。既存の変換定義は CLI によって自動的に処理されます。

変換レジストリ

変換レジストリは、変換定義を保存および管理するための AWS アカウントの一元化されたリポジトリです。レジストリ内の変換は次のとおりです。

- を使用して一覧表示 `atx custom def list`
- 複数のコードベースで実行
- AWS アカウントの他のユーザーと共有されている
- バージョン管理

⚠ Important

変換定義はアカウント固有です。別の AWS アカウントで変換を使用する場合は、そのアカウントで個別に公開する必要があります。

ドラフト変換と公開変換

AWS 変換カスタムは、レジストリ内の変換に 2 つの状態をサポートします。

ドラフト変換は、進行中またはテストされていない変換定義です。これらは特定のバージョンとして保存され、その特定のバージョンを参照するユーザーが取得、更新、実行できます。ドラフトは、変換をチームと共有できるようになる前に、反復的な開発、テスト、改良に役立ちます。ドラフトは特

定の会話にも関連付けられます。CLI を再起動する場合、`atx --conversation-id {id}`を使用して以前の CLI を復元できます。

公開された変換は、必要な IAM アクセス許可を持つ他のユーザーが実行できるように、アカウントの変換レジストリで使用できます。公開された変換は、を使用して検出できます`atx custom def list`。

一般的なワークフローは次のとおりです。

1. ローカルで変換を作成する
2. テスト用のドラフトとして保存 (`atx custom def save-draft`)
3. 絞り込みと検証
4. 発行してチームと共有します (`atx custom def publish`)

変換をドラフトとして保存せずに直接公開することもできます。

リファレンスと教訓

AWS 変換カスタムは、次の 2 種類の知識を使用して変換品質を向上させます。

リファレンスは、変換定義の `references`/フォルダに保存されているユーザー提供のドキュメントです。リファレンスはテキストファイルのみをサポートしており (すべてのファイルの合計は最大 10MB)、通常、ドキュメント、API 仕様、移行ガイド、コードサンプルが含まれています。リファレンスは、変換の実行中に必要に応じてロードされます。インタラクティブモードで変換定義を作成または更新するときに参照を追加します。

教訓は、変換の以前の実行から自動的に抽出されます。継続的な学習システムは、実行の軌跡、開発者のフィードバック、変換中に発生したコード修正からそれらを生成します。関連する教訓をカテゴリにグループ化し、まとめて確認できるようにします。事前に提供するリファレンスとは異なり、異なるコードベース間で変換を実行すると時間の経過とともに教訓が蓄積され、システムはそれらを自動的に適用して将来の実行を改善します。を使用して、教訓をインタラクティブに確認および管理します`atx custom def learnings`。

ビルドおよび検証コマンド

ビルドまたは検証コマンドは、変換プロセス中にコードを検証する方法を指定するオプションのパラメータです。このコマンドは、コードの整合性を確保するために、変換中のさまざまな時点で実行されます。

継続的な学習を通じて変換品質を向上させるには、結果を検証し、検証が失敗した場合に問題を返すコマンドを提供することが非常に重要です。ビルドまたは検証が必要ない場合は、入力から を省略します。

例と詳細なガイダンスについては、「ワークフロー」セクションの「ビルドおよび検証コマンド」を参照してください。

継続的な学習

継続的な学習は、すべての変換実行からのフィードバックを自動的にキャプチャし、時間の経過とともに変換品質を向上させるシステムです。システムは以下を通じて情報を収集します。

- 明示的なフィードバック - インタラクティブモードで提供されるコメントとコードの修正
- 暗黙的な観測 - コードの変換とデバッグ中にエージェントが遭遇する問題

継続的な学習システムはこの情報を処理して、将来の変換を改善するために変換定義に追加する教訓を作成します。変換が完了すると、システムは学習プロセスを自動的に実行するため、追加の入力は必要ありません。

Important

教訓は変換固有であり、異なる変換や異なる顧客アカウント間で共有されることはありません。

教訓の管理

システムは教訓を自動的に生成しますが、今後の実行に適用される教訓を制御できます。atx custom def learnings コマンドは、変換が蓄積した教訓を確認してキュレートするためのインタラクティブなターミナルセッションを開きます。

このセッションでは、次のことができます。

- カテゴリ別にグループ化された教訓を参照し、各カテゴリに含まれるアクティブな教訓の数を確認します。
- 教訓を開いて、教訓本文、その影響、それを参照した以前の実行数などの詳細をすべてお読みください。
- システムが今後の実行に適用できないように教訓をアーカイブし、必要に応じて後で復元します。

- 不要になったアーカイブされた教訓を完全に削除します。

手順については、「[the section called “継続的な学習”](#)」を参照してください。

クライアント側のスキル

クライアント側のスキルは、変換の実行中にエージェントを拡張する追加機能です。実行する変換を定義する変換定義とは異なり、クライアント側のスキルは、変換中にエージェントが組み込み機能と一緒に使用できる補足ツール、スクリプト、および手順を提供します。

クライアント側のスキルは、プロジェクトレベルおよびユーザーレベルのディレクトリから検出されます。リポジトリに固有の標準を適用する場合はプロジェクトレベルに、すべてのプロジェクトに適用される場合はユーザーレベルでスキルを配置します。プロジェクトレベルのスキルは、コードリポジトリパスが指定されている場合にのみ使用できます。詳細な設定と使用状況については、「」を参照してください[the section called “クライアント側のスキル”](#)。

開始方法

このセクションでは、AWS 変換カスタムをセットアップし、最初の変換を実行する方法について説明します。

前提条件

AWS Transform custom をインストールする前に、以下があることを確認してください。

サポートされているプラットフォーム

AWS 変換カスタムは、次のオペレーティングシステムをサポートしています。

- Linux - すべての Linux ディストリビューションのフルサポート
- macOS - macOS システムのフルサポート
- Windows - Windows ターミナルと PowerShell を使用したネイティブ Windows のフルサポート (atx ct コマンドはまだ Windows ではサポートされていません)
- Linux 用 Windows サブシステム (WSL) - WSL で実行されている場合にサポートされます

必要なソフトウェア

- Node.js 22 以降 - <https://nodejs.org/en/download> からダウンロード

- Git - 変換を実行するには、インストールする必要があり、作業ディレクトリが有効な Git リポジトリである必要があります。を実行して `git init`; `git add .`; `git commit -m "Initial commit"`、作業ディレクトリで Git リポジトリを初期化します。

ネットワークの要件

次のエンドポイントにアクセスできるインターネット接続が必要です。

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

インターネットが制限された環境で作業している場合は、ファイアウォールルールを更新してこれらの URLs。

Transform CLI AWS のインストール

推奨されるインストール方法は、インストールスクリプトを使用することです。オペレーティングシステムのタブを選択します。

AWS Transform CLI をインストールするには

1. インストールスクリプトを実行します。

Linux and macOS

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

Windows (PowerShell)

ネイティブ Windows では、PowerShell を使用して Windows ターミナルからインストールスクリプトを実行します。

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

2. インストールを確認します。

```
atx --version
```

コマンドには、インストールされているバージョンの AWS Transform CLI が表示されます。

認証の設定

AWS 変換カスタムには、サービスで認証するための AWS 認証情報が必要です。次のいずれかの方法を使用して認証を設定します。

環境可変

次の環境変数を設定します。

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"
```

AWS_PROFILE 環境変数を使用してプロファイルを指定することもできます。

Linux and macOS

```
export AWS_PROFILE=your_profile_name
```

Windows (PowerShell)

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

Note

PowerShell `$env:` の で設定された環境変数は、現在のセッションにのみ適用されます。セッション間で永続化するには、システムプロパティ > 環境変数で設定するか、 `[System.Environment]::SetEnvironmentVariable()` を使用します。

AWS 認証情報ファイル

Linux および macOS、または Windows `%USERPROFILE%\aws\credentials` のにある AWS 認証情報ファイル `~/.aws/credentials` で認証情報を設定します。

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

IAM 許可

AWS 認証情報には、AWS 変換カスタムサービスを呼び出すためのアクセス許可が必要です。 [AWSTransformCustomFullAccess](#) AWS 管理ポリシーを IAM ユーザーまたはロールにアタッチすることをお勧めします。このポリシーは、アカウントへの CloudWatch メトリクスの [放出に必要なサービスにリンクされたロール](#) を作成するアクセス許可を含む、AWS 変換カスタムへのフルアクセスを提供します。

より詳細な制御のために、AWS Transform custom には以下の AWS 管理ポリシーも用意されています。

- [AWSTransformCustomExecuteTransformations](#) – 変換を実行するためのアクセスを提供します。
- [AWSTransformCustomManageTransformations](#) – 変換リソースを作成、更新、読み取り、削除したり、変換を実行したりするためのアクセスを提供します。

これらのポリシーの詳細については、[AWS「Transform のマネージドポリシー」](#) を参照してください。リソースレベルのアクセス許可を持つカスタム IAM ポリシーについては、「カスタム [AWS IAM サービス認可の変換リファレンスガイド](#)」を参照してください。

 Note

- 変換定義は、ARNs AWS リソースネーム) を持つリソースです。IAM ポリシーを使用して、IAM ポリシーリソースステートメントで変換 ARN を指定することで、特定の変換または変換グループへのアクセスを制御できます。タグは、グループ化されたアクセスコントロールに使用できます。
- AWS AWS 変換にアクセスするには IAM アイデンティティセンターが必要ですが、CLI を使用する必要はありません。

AWS リージョンの設定

AWS 変換カスタムは特定の AWS リージョンで使用でき、標準の AWS CLI 優先順位を使用してリージョン設定を自動的に検出します。

サポート対象のリージョン

AWS 変換カスタムは、次の AWS リージョンで使用できます。

- us-east-1 (米国東部 - バージニア北部)
- eu-central-1 (欧州 - フランクフルト)
- eu-west-2 (欧州 - ロンドン)
- ca-central-1 (カナダ - 中央)
- ap-northeast-1 (アジアパシフィック - 東京)
- ap-northeast-2 (アジアパシフィック - ソウル)
- ap-southeast-2 (アジアパシフィック - シドニー)
- ap-south-1 (アジアパシフィック - ムンバイ)

リージョンの決定方法

CLI は、これらのソースを優先度順にチェックして、使用するリージョンを決定します。

1. AWS_REGION 環境変数 (最高優先度)
2. AWS_DEFAULT_REGION 環境変数
3. AWS 設定ファイルで選択したプロファイル (AWS_PROFILE環境変数経由)

4. AWS 設定ファイルのデフォルトプロファイル

5. 設定が見つからないus-east-1場合のデフォルトのフォールバック

Note

AWS 設定ファイルは、`~/.aws/config` Linux および macOS の場合は、Windows `%USERPROFILE%\aws\config` の場合は にあります。

Note

`ATX_CUSTOM_ENDPOINT` が設定されている場合、リージョンはエンドポイント URL から抽出され、他のすべての設定が上書きされます。

リージョンの設定

次のいずれかの方法を選択して、リージョンを設定します。

オプション 1: 環境変数 (一時的な使用に推奨):

Linux and macOS

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

Windows (PowerShell)

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

オプション 2: AWS CLI 設定 (永続的なセットアップに推奨):

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

オプション 3: プロファイル固有の設定:

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"
```

オプション 4: ファイルを直接編集する:

設定ファイルを直接編集します (~/.aws/configLinux AWS および macOS の場合、または Windows %USERPROFILE%\aws\config の場合)。

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

リージョン設定の検証

Transform AWS custom が使用するリージョンを確認するには:

現在のリージョン設定を確認します。

```
aws configure get region
```

環境変数を確認します。

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"
```

デバッグログで詳細なリージョン解決を表示します。

Linux and macOS

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

Windows (PowerShell)

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

リージョンソースを示すログ出力の例:

```
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 フィールドには、リージョンの派生元 (例: 「aws-config (プロファイル: default)」、AWS_REGION) が表示されます。

Important

リージョン設定がサポートされていないリージョンを指している場合、CLI は、サポートされているリージョンに設定を更新する手順を含む明確なエラーメッセージを表示します。

最初の変換の実行

開始する最も簡単な方法は、AWSマネージド変換を使用することです。AWSマネージド変換は、事前に構築され、AWS検証された変換であり、セットアップなしですぐに使用できます。

AWSマネージド変換をインタラクティブに実行するには:

```
atx
```

次に、変換を実行するようにエージェントに依頼します。

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

変換を非インタラクティブに実行するには:

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

実行モード、設定オプション、コマンドフラグの詳細については、「」を参照してください[the section called “コマンドリファレンス”](#)。

変換結果について

変換が完了すると、AWS Transform custom は結果を記述するレポートを提供します。変換が検証されなかった場合 (Java アプリケーションのビルドが成功した場合など)、レポートには今後のアクションに関する推奨事項が表示されます。

ほとんどの変換は複数のステップで実行され、Git を使用して中間変換をコミットします。変換によって行われた変更は、標準の Git コマンドを使用して確認できます。

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

カスタム変換の作成

AWS マネージド変換がニーズに合わない場合は、特定の要件に合わせたカスタム変換を作成できます。

カスタム変換を作成するには

1. インタラクティブモードで AWS 変換 CLI を起動します。

```
atx
```

2. 新しい変換を作成するエージェントに指示し、要件を記述します。
3. 品質を向上させるために、ドキュメントやコードサンプルなどの参考資料を提供します。
4. 変換を繰り返しテストして絞り込みます。
5. 組織の変換レジストリに公開します。

カスタム変換を作成する詳細な手順については、「」を参照してください[the section called “ワークフロー”](#)。

AWS ウェブアプリケーションの変換 (オプション)

AWS Transform ウェブアプリケーションは、複数のリポジトリで大規模な変換キャンペーンをモニタリングするためのオプションのインターフェイスです。AWS 変換ウェブアプリケーションを使用する前に、組織内の AWS 変換にアクセスするためのユーザー ID を有効にする必要があります。AWS 変換を有効にする方法については、[AWS 「変換の設定」](#)を参照してください。

AWS 変換ウェブアプリケーションを使用するには:

1. <https://aws.amazon.com/transform/> にアクセスして、IAM アイデンティティセンター AWS の認証情報を使用してサインインします。
2. AWS 変換ウェブアプリケーションで「カスタム/コード」ジョブタイプを選択します。
3. 使用する変換定義の名前をエージェントに伝えます。
4. ウェブアプリケーションは、チームと共有できる AWS Transform CLI コマンドを提供します。このコマンドは変換定義を呼び出し、実行結果をウェブアプリケーションに記録します。
5. 変換結果がキャプチャされたら、統計情報、時間節約の測定値を表示し、Dashboard でジョブの進行状況について質問します。

ウェブアプリケーションは、複数のコードベースにわたる変換を一元的にモニタリングする必要があるエンタープライズ規模のオペレーション向けに設計されています。

Note

AWS AWS Transform ウェブアプリケーションにアクセスするには、IAM Identity Center が必要です。CLI には IAM Identity Center は必要ありません。標準の AWS 認証情報のみが必要です。

カスタム変換コマンドの概要

カスタム変換で使えるコマンドをいくつか示します。コマンドの完全なリストは、[AWS 「変換カスタム変換コマンドリファレンス」](#)にあります。

- `atx custom`

- カスタム変換インタラクティブエクスペリエンスを実行し、変換の作成、検出、実行、および改良を可能にします。
- これは のデフォルトのコマンドです `atx`。
- `--trust-all-tools (-t)` はオプションであり、エージェントによってリクエストされたすべてのツールリクエストを暗黙的に許可します。特に本番環境では注意が必要です。信頼[設定ファイル](#)を使用して、特定のツールとコマンドのツール信頼を設定できます。
- `atx custom --help` | `atx custom -h`
 - ヘルプメニューを表示します。
 - 各コマンドには、 などのヘルプメニューも含まれています `atx custom def exec --help`。
- `atx --version` | `atx -v`
 - バージョンを表示します。
 - バージョン番号はリリースごとに変わります。
- `atx custom def list`
 - 変換レジストリで利用できる変換のリストを出力します。
- `atx custom def exec`
 - 変換を実行します。
- `atx custom def learnings`
 - インタラクティブセッションを開き、変換が以前の実行から学んだ教訓を表示および管理します
- `atx mcp`
 - MCP サーバー設定の管理に使用されます

ワークフロー

変換の実行

このセクションでは、変換を実行するさまざまな方法と、実行動作を制御するためのオプションについて説明します。

実行モード

AWS 変換カスタムは、異なるワークフローに対応するために 3 つの実行モードをサポートしています。

インタラクティブ対話モード

で CLI を起動atxし、自然言語による変換を実行するようにエージェントに依頼します。このモードでは、エージェントと完全に会話し、いつでも実行を中断し、変換プロセス中にフィードバックを提供できます。

このモードは、最大限の制御と、複雑なシナリオでエージェントをガイドする機能が必要な場合に使用します。

直接インタラクティブ実行

`atx custom def exec -n <transformation-name> -p <path>` を使用して、特定の変換をインタラクティブに開始します。このモードでは、エージェントを最初、最中、または実行終了時に確認して操作できます。エージェントは主要な決定ポイントで一時停止し、入力を求めます。

これは、トランスフォーメーションを自律的に実行する前にテストおよび改良するのに最適です。

非インタラクティブモードまたはヘッドレスモードで変換を実行できます。非インタラクティブモードでは、名前付き変換中のプロンプトが抑制されます。ヘッドレスモードでは、インタラクティブインターフェイスを完全にバイパスして、プレーンテキストのプロンプトでエージェントを実行できます。

非インタラクティブモード

完全な自動化atx `atx custom def exec -n <transformation-name> -p <path> -x -t`にはを使用します。`-x` を非インタラクティブモードで実行し、`-t` を追加してプロンプトを表示せずにすべてのツールを自動的に信頼します。

このモードは、人間による介入が利用できない、または望ましくない CI/CD パイプライン統合と一括実行用に設計されています。

ヘッドレスモード

エージェントとやり取りせずにタスクを完了するには、`atx -x "<prompt>" -t`、プレーンテキストで指示を提供します。

ヘッドレス変換の実行

このモードを使用して、既存の変換定義をコードベースに適用します。変換は、承認を必要とせずに各ステップを自動的に実行します。

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

ヘッドレス変換の開発

変換定義を作成または変更します。

レガシー変換定義を新しいスキル形式 (SKILL.md + references/) に変換するには、次のコマンドを実行します。

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

新しい変換定義を作成するには、次のコマンドを実行します。

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

一般的なコマンドフラグ

で変換を実行する場合 `atx custom def exec`、一般的に次のフラグが使用されます。

- `-n` または `--transformation-name` - 実行する変換の名前を指定します。
- `-p` または `--code-repository-path` - コードベースへのパスを指定します (現在のディレクトリには「`.`」を使用します)
- `-c` または `--build-command` - 実行するビルドまたは検証コマンドを指定します
- `-x` または `--non-interactive` - 非インタラクティブモードを有効にします (ユーザープロンプトなし)
- `-t` または `--trust-all-tools` - プロンプトを表示せずにすべてのツールを自動的に信頼します
- `-d` または `--do-not-learn` - この実行からの教訓の抽出を防止します
- `--tv` または `--transformation-version` - 変換の特定のバージョンを指定します。
- `-g` または `--configuration` - 設定ファイルまたはインライン設定を提供します

Important

`-t` または `--trust-all-tools` フラグは、プロンプトを表示せずにすべてのツール実行を自動的に承認し、ほとんどのセキュリティガードレールをバイパスします (によって上書きされない限り、`alwaysPromptCommands` リストに一致するコマンドには明示的なアクセス許可が必要です `trustedShellCommands`)。完全に自律したエクスペリエンスには `--non-`

`interactive`と `を渡す--trust-all-tools`必要がありますが、変換を実行するためには必要ありません。本番稼働環境での使用には注意が必要です。

設定ファイルの使用

AWS 変換カスタムは、YAML または JSON 形式のオプションの設定ファイルをサポートしています。設定ファイルを使用すると、実行パラメータを指定し、エージェントに追加のコンテキストを提供できます。

設定ファイルを使用するには:

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

設定をインラインキーと値のペアとして指定することもできます。

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

設定ファイルの例 (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` パラメータは、エージェントの実行プランに追加のコンテキストを提供します。これは、AWS マネージド変換で特定のニーズに合わせて動作をカスタマイズするために特に役立ちます。

ビルドおよび検証コマンド

ビルドまたは検証コマンドは、変換プロセス中にコードを検証する方法を指定するオプションのパラメータです。AWS Transform custom は、指定しない場合、変換に基づいて最適なビルドコマンドを推測しようとはしますが、品質に固有であることが推奨されます。

ビルドコマンドと検証コマンドの例:

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

構築を必要としない言語や変換であっても、結果を検証し、検証が失敗した場合に問題を返すコマンドを提供することは、変換の品質を向上させるために非常に重要です。

ビルドまたは検証が必要ない場合は、入力から `build` を省略します。

学習動作の制御

デフォルトでは、AWS Transform custom はすべての変換実行から教訓を抽出します。特定の実行の学習を防ぐことができます。

実行からの学習を防ぐには:

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

`-d` または `--do-not-learn` フラグは、現在の実行からの教訓抽出の許可をオプトアウトします。

会話を再開する

AWS 変換カスタムを使用すると、作成から 30 日以内に以前の会話を再開できます。

最新の会話を再開するには:

```
atx --resume
```

特定の会話を再開するには:

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

Important

会話は、作成後 30 日以内にのみ再開できます。30 日後、会話を再開できなくなります。

エージェント分の追跡

AWS 変換カスタムは、変換セッション中に消費された[エージェントの分](#)を追跡します。エージェントの分は会話のライフサイクル全体で累積され、会話が終了すると表示されます。

```
Agent minutes used: 12.50
```

エージェント分は中断後も保持されます。Ctrl+C を使用してセッションを中断し、後で再開すると、以前に蓄積された分が引き継がれ、再開されたセッションで蓄積し続けます。

インタラクティブセッション中にエージェント分を確認するには:

入力プロンプト/usageに と入力して、会話を終了せずに現在蓄積されているエージェント分を表示します。

エージェント分予算制限を設定するには:

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

--limit オプションは、セッションの[エージェント分](#)予算の上限を設定します。エージェント分は、ウォールクロック時間ではなく、アクティブなエージェントの作業時間を反映します。制限に達すると、CLI はメッセージを表示し、再開する手順で終了します。

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

制限を引き上げて後で会話を再開できます。

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

継続的な学習

このセクションでは、継続的学習によって作成された教訓を確認して管理する方法について説明します。

教訓を理解する

継続的な学習システムは、変換の以前の実行から教訓を自動的に抽出します。システムは、以下に基づいて非同期的にそれらを作成します。

- インタラクティブモードで提供される開発者フィードバック
- 変換中に発生したコードの問題

異なるコードベース間で変換を実行すると、時間の経過とともに教訓が蓄積されます。システムはそれらを自動的に適用して、今後の実行を改善します。各教訓は、関連する教訓と一緒に確認できるように、類似ドメインのすべての教訓を含むカテゴリに属します。使用したくない教訓については、その教訓を完全にアーカイブまたは削除できます。

教訓の表示と管理

`learnings` コマンドを使用して、変換定義の教訓を参照および管理するためのインタラクティブセッションを開きます。

教訓ビューワーを開くには:

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

ビューワーは、教訓カテゴリのリストで開き、それぞれにアクティブな教訓の数が表示されます。カテゴリを選択してその教訓を表示し、次に教訓を選択して、教訓本文、その影響、それを参照した以前の実行の数など、その詳細全体を表示します。

教訓のアーカイブと復元

システムは教訓を自動的に適用します。システムに教訓を適用させたくない場合は、その教訓をアーカイブできます。アーカイブされた教訓は保持されますが、今後の実行には適用されません。アーカイブされたすべての教訓はグループ化されているため、それらを確認してアクティブな使用に復元できます。

教訓の削除

役に立たない教訓を完全に削除します。削除を元に戻すことはできず、システムは今後の実行から削除された教訓を再学習する可能性があります。

教訓は、削除する前にアーカイブする必要があります。

高度な設定

このセクションでは、AWS 変換カスタムの高度な機能と設定オプションについて説明します。

環境可変

環境変数を使用して CLI の動作をカスタマイズできます。

Note

次の例は、Linux および macOS 構文 () を示しています `export`。Windows では、 `Set-EnvironmentVariable` を使用して PowerShell で環境変数を設定します `$env:NAME="value"`。同等のコマンドについては、Windows (PowerShell) タブを参照してください。

ATX_SHELL_TIMEOUT

シェルコマンドのデフォルトのタイムアウトを上書きします (900 秒/15 分)。

Linux and macOS

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

Windows (PowerShell)

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

これは、大規模なコードベースや長時間実行されるビルドプロセスに役立ちます。

ATX_DISABLE_UPDATE_CHECK

コマンドの実行中に自動バージョンチェックを無効にし、通知を更新します。

Linux and macOS

```
export ATX_DISABLE_UPDATE_CHECK=true
```

Windows (PowerShell)

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

ATX_GIT_COMMITTER_NAME および ATX_GIT_COMMITTER_EMAIL

Transform がリポジトリに作成するチェックポイントコミットに使用される作成者 ID AWS を設定して、変換中に変更を適用します。これらの変数が設定されていない場合、チェックポイントコミットはデフォルトの ID () に属性付けられます ATX Bot <checkpoint@atx.bot>。チェックポイントを特定の作成者に属性するように両方の変数を設定します。

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"
```

信頼設定

信頼設定を使用すると、プロンプトなしで実行する特定のツールやコマンドを事前に承認できます。信頼レベルに関係なく、特定のシェルコマンドに明示的なアクセス許可を要求することもできます。これらの設定は ~/.aws/atx/trust-settings.yaml ファイルで行われます。

ファイルには 3 つのリストが含まれています。

- trustedTools - プロンプトを表示せずに実行できるツール
- trustedShellCommands - プロンプトを表示せずに実行できるシェルコマンド
- alwaysPromptCommands - -t フラグまたはセッションの信頼に関係なく、によって上書きされない限り trustedShellCommands、明示的なアクセス許可を必要とするシェルコマンドパターン。これらのパターンは、非インタラクティブモード () では強制されません -x。

デフォルトの信頼されたツール:

- file_read
- get_transformation_from_registry
- list_available_transformations_from_registry

信頼設定の編集:

trust-settings.yaml ファイルを手動で編集して、信頼できるツールやコマンドを追加または削除できます。trustedShellCommands とはどちらも、を使用した glob ワイルドカードパターン alwaysPromptCommands をサポートしています*。

 Note

コマンドが両方のリストと一致する場合、が優先 trustedShellCommands されます。

以下に、各コマンドリストと例を示します。

- trustedShellCommands - これらのパターンに一致するコマンドは、プロンプトを表示せずに実行され、他のすべてのガードレールをバイパスします。パターンは完全なコマンド文字列と照合されます。

例:

- cd * - cd で始まる複合コマンドに一致
- *&&* - && 演算子を使用してすべてのコマンドを信頼します
- alwaysPromptCommands - これらのパターンに一致するコマンドには、-t フラグまたはセッションの信頼に関係なく、によって上書きされない限り trustedShellCommands、明示的なアクセス許可が必要です。これらのパターンは、非インタラクティブモード () では強制されません-x。パターンは、複合式 (&&、||、コマンド置換) の各サブコマンドと照合されます。

例:

- rm -rf * - 再帰的な強制削除コマンドを常に求める
- sudo * - sudo で実行されるコマンドを常にプロンプトする
- find * -exec * - 常に -exec で検索コマンドを要求する

セッションレベルの信頼:

インタラクティブプロンプトでは、以下を選択できます。

- (y)es - 1 回実行する
- (n)o - 拒否
- (t)rust - 現在のセッションのみを信頼する

セッションレベルの信頼設定は一時的なもので、CLI の再起動時にリセットされるため、`trust-settings.yaml` を完全に変更することなく一時的な承認が得られます。

Note

セッション信頼は、`alwaysPromptCommands` リストに一致するコマンドでは使用できません。

モデルコンテキストプロトコル (MCP) サーバー

AWS Transform CLI は、モデルコンテキストプロトコル (MCP) サーバーをサポートしています。MCP サーバーは、追加のツールで機能を拡張します。

設定:

`~/.aws/atx/mcp.json` ファイルで MCP サーバーを設定します。AWS Transform CLI は、ローカルコマンドベースのサーバーとリモート HTTP サーバーの 2 種類の MCP サーバーをサポートしています。

ローカルコマンドベースのサーバー:

ローカルサーバーは、マシンで子プロセスとして実行されます。`command` プロパティを使用して設定します。

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

リモート HTTP サーバー:

リモートサーバーは、HTTP または HTTPS URL でホストされている MCP サーバーに接続します。`url` プロパティを使用して設定します。

```
{
```

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

headers プロパティはオプションで、`${VAR_NAME}` 構文を使用した環境変数の拡張をサポートしています。これにより、API トークンなどの機密値を設定ファイルではなく環境変数に保存できます。

設定プロパティ:

ローカルコマンドベースのサーバーは、次のプロパティをサポートしています。

- `command` (必須) - サーバーを実行するコマンド
- `args` (オプション) - コマンドライン引数の配列
- `env` (オプション) - サーバードメインに渡す環境変数

リモート HTTP サーバーは、次のプロパティをサポートしています。

- `url` (必須) - リモート MCP サーバーの HTTP または HTTPS URL
- `headers` (オプション) - `${VAR_NAME}` 環境変数の拡張をサポートするリクエストに含める HTTP ヘッダー

MCP サーバーの管理:

設定された MCP サーバーのリストを表示します。

```
atx mcp tools
```

特定の MCP サーバーが提供する利用可能なツールを一覧表示します。

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

使用状況の追跡:

CLI は、変換の実行中に MCP ツールの使用を自動的に追跡します。使用状況統計は、とともに会話ディレクトリ `mcp_usage.json` のとして保持されます `metadata.json`。ファイルには、以下を含む各実行のツールごとのメトリクスが記録されます。

- ツールあたりの呼び出し数
- ツールあたりのエラー数
- ツールあたりの合計実行時間
- 最後のエラーの詳細 (存在する場合)

クライアント側のスキル

クライアント側のスキルは、変換の実行中にエージェントを拡張する追加機能です。これにより、エージェントが組み込み機能とともに使用できるカスタムツール、スクリプト、および手順を提供できます。

スキル検出ディレクトリ:

スキルは、4 つのディレクトリから優先順位で検出されます。同じ名前のスキルが複数のディレクトリに存在する場合、リストの最初のディレクトリが優先されます。

1. `<project>/aws/atx/skills/` - プロジェクトレベル、AWS 変換 CLI 固有
2. `<project>/agents/skills/` - プロジェクトレベル、クロスクライアント (互換性のある任意のエージェントツールで使用可能)
3. `~/aws/atx/skills/` - ユーザーレベル、CLI 固有の AWS 変換
4. `~/agents/skills/` - ユーザーレベル、クロスクライアント (互換性のある任意のエージェントツールで使用可能)

`aws/atx/skills/` ディレクトリは AWS Transform CLI に固有です。`agents/skills/` ディレクトリはクロスクライアントです。つまり、そこに配置されたスキルは、AWS Transform CLI 以外の互換性のあるエージェントツールで使用できます。

スキルディレクトリ構造:

各スキルは、YAML フロントマターを持つ `SKILL.md` ファイルを含むディレクトリです。

```
~/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 形式:

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

Instructions for the agent...
```

name フィールドは親ディレクトリ名と一致する必要があります。

スキルの無効化:

ファイルを削除せずにスキルがロードされないようにするには、`disable-model-invocation: true` をフロントマターに追加します。

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

このプロパティを設定すると、CLI は検出中にスキルをスキップします。変換定義がスキルファイルを読み取るように明示的に指示しない限り、エージェントはスキルを表示または使用することはできません。これを使用して、スキルを一時的に無効にしたり、work-in-progressとしてマークしたり、人間の読者のみを対象とした参考資料を保持したりできます。

Note

無効なスキルのファイルはディスクに残ります。変換定義によって特定のファイルパスを読み取るようにエージェントに指示された場合でも、エージェントはコンテンツにアクセスできます。disable-model-invocation プロパティは、ファイルシステムアクセスではなく、自動検出とコンテキストインジェクションを防ぎます。

実行モード別のスキルの可用性:

- Exec モード (atx custom def exec と --code-repository-path) - ユーザーレベルとプロジェクトレベルの両方のディレクトリからスキルを検出します。
- インタラクティブモード (atx) - ユーザーレベルのスキルのみが最初に検出されます。セッション中にコードリポジトリパスを指定すると、プロジェクトレベルのスキルもロードされます。

スキル検出の検証:

実行後に CLI のデバッグログをチェックして、検出されたスキルを確認します。

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
```

検証に失敗したスキルは、デバッグログに警告とともにスキップされます。

Note

クライアント側のスキルには CLI バージョン 2.0 以降が必要です。

プロジェクトレベルとユーザーレベルのスキルの選択

スキルを配置する場所によって、そのスキルの利点とアクティブ化時期が決まります。

プロジェクトレベルのスキル (<project>/.aws/atx/skills/):

これらをバージョン管理にコミットして、リポジトリに対して変換を実行しているすべてのチームメンバーがそれらを自動的に検出できるようにします。以下のプロジェクトレベルのスキルを使用します。

- リポジトリ固有のコンプライアンスチェック (Dockerfile ルール、Terraform ポリシー、移行安全バリデータ)

- このコードベースに適用される組織のコーディング標準 (オブザーバビリティパターン、エラー処理、命名規則)
- プロジェクトに固有のスクリプトを構築またはテストする (カスタム linters、アーキテクチャのフィットネス機能)
- このリポジトリで使用する内部ライブラリの API 移行ガイド

ユーザーレベルのスキル (~/.aws/atx/skills/):

これらはマシンに残り、ターゲットとするリポジトリに関係なく、すべての変換中にアクティブ化されます。ユーザーレベルのスキルは、以下に使用します。

- 個人用ワークフローツール (変更ログジェネレーター、コミットメッセージフォーマッター)
- クロスプロジェクト設定 (推奨テストパターン、ドキュメントスタイルのリマインダー)
- すべてのリポジトリで組織が必要とするライセンスコンプライアンスチェック
- 使用するすべてのコードベースに適用されるカバレッジしきい値または品質ゲート

効果的なスキルのヒント:

- SKILL.md フロントマターにクリアdescriptionフィールドを書き込みます。エージェントはこのフィールドを使用して、スキルがいつ関連しているかを決定します。
- 成功の場合はコード 0、失敗の場合はゼロ以外の検証スクリプトを終了します。エージェントは終了コードを解釈してコンプライアンスを判断します。
- 明確で実用的なエラーメッセージをスクリプトに出力します。エージェントは出力を読み取り、修正する内容を理解します。
- いずれかのレベルでクロスクライアントディレクトリ (.agents/skills/) にスキルを配置し、AWS Transform CLI 以外の他の AI 開発ツールと共有します。

クライアント側のスキルの例

これらの例は、スクリプトベースの検証スキルと参照のみのスキルの 2 つの一般的なパターンを示しています。

例: Dockerfile Compliance Checker (スクリプトベース)

このスキルは、セキュリティと運用のベストプラクティスに照らして Dockerfiles を検証します。変更の前後にエージェントが実行する検証スクリプトを使用します。

ディレクトリ構造:

```
.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.
```

検証スクリプトは、ピン留めされていないベースイメージタグ、ルートとして実行されているタグ、ENVディレクティブでハードコードされたシークレット、HEALTHCHECK定義の欠落をチェックします。エージェントはスクリプトを実行し、参照ファイルのパターンを使用して違反を修正し、スクリプトを再実行してコンプライアンスを確認します。

例: API 非推奨ヘルパー (参照のみ)

このスキルは、アップグレード変換中に非推奨の API コールを置き換えることを通じてエージェントをガイドします。スクリプトのない参照ファイルのみを使用します。

ディレクトリ構造:

```
.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
```

リファレンスファイルにはbefore-and-afterコード例が含まれています。例えば、は `S3Client`と を使用して、 のようなパターン `s3.putObject(params).promise()` をモジュール v3 に相当する に `aws-sdk-v2-to-v3.md` マッピングします `PutObjectCommand`。

タグと組織

アクセスコントロールと分類のタグを使用して変換を整理できます。

Note

これらのコマンドの一部では、変換定義の Amazon リソースネーム (ARN) を指定する必要があります。ARN 構造は次のとおりです。arn:aws:transform-custom:<region>:<account-id>:package/<td-name>

変換のタグを一覧表示するには:

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

変換にタグを追加するには:

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

変換からタグを削除するには:

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

タグは、IAM ポリシーのグループ化されたアクセスコントロールに使用できます。特定のタグを持つすべての変換 (例: team:frontendまたは environment:production) にアクセス許可を付与するポリシーを作成できます。

ログ

AWS Transform CLI は、トラブルシューティングとデバッグのために 3 種類のログを保持します。

会話ログ:

Linux and macOS

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

Windows

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

これらのログには、特定のセッションの完全な会話履歴が含まれます。

サブエージェントログ:

Linux and macOS

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

Windows

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

これらのログには、変換中にメインエージェントがスポンするサブエージェントからの出力が含まれています。サブエージェントを直接管理する必要はありません。

開発者デバッグログ:

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
```

これらのログは、CLI 自体の高度なトラブルシューティング情報を提供します。

Note

Logs ディレクトリに複数のデバッグログファイルがある場合があります (debug1.log、debug2.log など)。サポートチケットを開いてより迅速に解決する場合は、`~/.aws/atx/custom/<conversation-id>/*` や `~/.aws/atx/logs/*` など、関連するすべてのログを確認して提供します。

CLI 更新

CLI を最新の状態に保つことで、新機能や改善点にアクセスできます。

更新を確認するには:

```
atx update --check
```

を最新バージョンに更新するには:

```
atx update
```

特定のバージョンに更新するには:

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

カスタム変換を作成する

このセクションでは、カスタム変換定義を作成、変更、管理する方法について説明します。

新しい変換の作成

インタラクティブ CLI を使用して、新しい変換定義を作成します。

変換定義を作成するには

1. AWS 変換 CLI を起動します。

```
atx
```

2. 新しい変換を作成することをエージェントに伝えます。
3. 変換目標の明確で詳細な説明を入力します。以下が含まれます:
 - ソースとターゲットの状態 (例: 「バージョン X からバージョン Y へのアップグレード」)
 - 必要な特定の変更 (「インポートステートメントの更新、廃止されたメソッドの置き換え」など)
 - 特別な考慮事項または制約事項
4. エージェントが明確化や追加情報をリクエストする場合は、具体的な例と参考資料を提供します。
5. エージェントによって作成された最初の変換定義を確認します。
6. サンプルコードベースで変換をテストします。

7. フィードバック、コード修正、または追加の例を提供して反復します。
8. 変換をローカルに保存するか、レジストリに公開します。

変換を作成するためのベストプラクティス:

- 複雑な変換を試みる前に、シンプルで明確に定義された変換から始める
- 移行ガイドやコードサンプルを含む包括的な参考資料を提供する
- 公開前に複数のサンプルコードベースでテストする
- 決定論的なビルドコマンドまたは検証コマンドを使用して継続的な学習を有効にする
- 複雑な変換を複数の小さなステップに分割することを検討する
- エージェントがこれらの要件を優先するように、変換定義に「CRITICAL:」または「IMPORTANT:」で重要な情報をマークします。
- 正確な要件に従う必要がある場合 (特定のコマンドや文字列値を使用するなど)、変換定義で完全な文字列を明示的に指定します。これらをバッシュ引用符で囲んで、それらがターミナルコマンドまたはリテラル文字列であることを明確に示すことができます。これにより、ばらつきが軽減され、一貫した実行が保証されます。

参考資料の提供

会話中にファイルパスを指定することで、AWS 変換カスタムにリファレンスファイルを提供できます。これらのファイルは、変換定義の `references/` フォルダに保存されます。

推奨されるリファレンスファイルのタイプ:

- 前後のサンプルコード
- 関連する APIs、ライブラリ、または機能のドキュメント
- 人間が読める移行ガイド

リファレンスファイルを指定するには:

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

複数の参照ファイルを含むディレクトリを指定することもできます。

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

Note

テキストベースのファイル (.md、.html、.txt、コードファイル) のみがサポートされています。バイナリファイル、イメージ、リッチテキストファイル (.pdf、.png、.docx など) は現在サポートされていません。多くの場合、テキストコンテンツを抽出し、それをリファレンスとして使用できます。小さなテキストファイルが多数ある場合は、記述名付きファイルに連結することを検討してください。すべてのファイルの合計は 10MB に制限されています。

既存の変換の変更

カスタム変換は、ドラフトとして保存するか公開する前と後に変更できます。AWS マネージド変換は変更できません。カスタマイズする必要がある場合は、設定ファイルを使用して追加のコンテキストを指定できます。

既存の変換を変更するには

1. AWS 変換 CLI を起動します。

```
atx
```

2. 既存の変換を変更することをエージェントに伝えます。
3. 以下を行うかどうかを選択します。
 - ローカルに保存された変換へのファイルパスを指定する (つまり、保存されたドラフトや公開されていない)
 - レジストリから変換のリストをリクエストする
4. レジストリから選択する場合は、変更する変換を選択します。
5. エージェントと協力して、行う変更を記述します。
6. サンプルコードベースで更新された変換をテストします。
7. 必要に応じて、更新をレジストリに発行します。

変換の発行と管理

インタラクティブエクスペリエンスまたは次のコマンドを使用して、変換を発行および管理できます。

変換をドラフトとして保存するには:

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

変換を公開するには:

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

使用可能な変換を一覧表示するには:

```
atx custom def list
```

変換定義をダウンロードするには:

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

これにより、変換定義が現在の作業ディレクトリにダウンロードされます。ターゲットディレクトリは `--td` フラグで指定でき、バージョンは `--tv` フラグで指定できます。

変換定義を削除するには:

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

Important

これにより、指定された変換定義がアカウントから完全に削除されます。

変換バージョンの管理

AWS 変換カスタムは、変換定義のバージョンを維持します。変換を実行またはダウンロードするときにバージョンを指定できます。

特定のバージョンを実行するには:

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

特定のバージョンをダウンロードするには:

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

バージョンが指定されていない場合は、最新バージョンが使用されます。

コマンドリファレンス

このセクションでは、すべての AWS 変換カスタム CLI コマンドの包括的なリファレンスを提供します。

インタラクティブコマンド

atx

AWS Transform CLI を使用してインタラクティブな会話を開始します。

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

インタラクティブセッションコマンド

次のコマンドは、インタラクティブセッション中に入力プロンプトで入力できます。

- /usage - 現在のセッションの累積[エージェント分](#)を表示します。

変換定義コマンド

atx custom def exec

コードリポジトリで変換定義を実行します。

オプション	ロングフォーム	パラメータ	説明
-p	--code-repository-path	<path>	変換するコードリポジトリへのパス。現在のディレクトリには、「」を使用します。

オプション	ロングフォーム	パラメータ	説明
-c	--build-command	<command>	リポジトリの構築時に実行するコマンド
-n	--transformation-name	<name>	レジストリ内の変換定義の名前
-x	--non-interactive	-	ユーザー支援なしで変換を実行します
-t	--trust-all-tools	-	すべてのツールを信頼します (ツールプロンプトなし)
-d	--do-not-learn	-	この実行からの教訓抽出の許可をオプトアウトする
-g	--configuration	<config>	設定ファイル (JSON または YAML) または key=value ペアへのパス
--tv	--transformation-version	<version>	使用する変換定義のバージョン
--limit	--limit	<limit>	エージェント分 予算制限を設定します。制限に達すると変換が終了し、制限を引き上げて再開できます
-h	--help	-	コマンドのヘルプを表示する

atx カスタム定義リスト

使用可能な変換定義を一覧表示します。

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

atx カスタム定義の取得

特定の変換定義の詳細を取得します。

オプション	ロングフォーム	パラメータ	説明
-n	--transfo rmation-name	<name>	取得する変換定義の 名前
--tv	--transfo rmation-v ersion	<version>	取得する変換定義の バージョン
--td	--target- directory	<directory>	変換定義を保存する ターゲットディレク トリ (デフォルト: ".")
--json	--json	-	JSON 形式でレスポ ンスを出力する
-h	--help	-	コマンドのヘルプを 表示する

atx カスタム定義の削除

変換定義を完全に削除します。

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

atx カスタム定義の公開

変換定義をレジストリに発行します。

オプション	ロングフォーム	パラメータ	説明
-n	--transformation-name	<name>	発行する変換定義の名前
--tv	--transformation-version	<version>	使用する変換定義のバージョン (--description または --source-directory には適用されません)
--sd	--source-directory	<directory>	ローカル変換定義ファイルを読み取るソースディレクトリ (--transformation-version には適用されません)
--description	--description	<description>	変換定義の説明 (--transformation-version では該当しません)
--json	--json	-	JSON 形式でレスポンスを出力する
-h	--help	-	コマンドのヘルプを表示する

atx custom def save-draft

変換定義をドラフトとしてレジストリに保存します。ドラフトは 30 日後に期限切れになり、レジストリに表示されません。このコマンドは、ドラフトのバージョン番号を返します。変換名とバージョン番号を使用して、ドラフトを明示的に実行および取得する必要があります。

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


Lesson コマンド

atx カスタム def 学習

インタラクティブセッションを開いて、変換定義が以前の実行から学んだ教訓を表示および管理します。カテゴリ別に教訓を参照し、教訓の詳細を表示し、教訓をアーカイブ、復元、または削除します。

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

タグコマンド

atx カスタム定義リストタグ

変換定義のタグを一覧表示します。

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

atx カスタム定義タグ

変換定義にタグを追加します。

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

atx カスタム定義タグ解除

変換定義からタグを削除します。

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

コマンドの更新

atx 更新

AWS Transform CLI を更新します。

```
atx update          # Update to latest version
atx update --check   # Check for updates only
```

```
atx update --target-version <version> # Update to specific version
```

MCP コマンド

atx mcp ツール

クライアントが MCP ツール設定を表示/確認できるようにします。設定の更新は、`~/.aws/atx/mcp.json` ファイルを介して直接管理する必要があります

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

AWS マネージド変換

AWS マネージド変換は、追加のセットアップなしですぐに使用できる一般的なユースケース向けに事前に構築され、AWS 検証された変換です。

概要

AWS マネージド変換には次の特性があります。

- による検証 AWS - これらの変換は によって高品質 AWS であることが検証されます。
- すぐに使用可能 - 追加のセットアップは必要ありません
- 継続的な成長 - 追加の変換が継続的に追加されています
- カスタマイズ可能 - `additionalPlanContext` 設定パラメータを使用して、組織のニーズに固有の追加のガイダンスまたは要件を提供することで、構築済みの変換をカスタマイズできます。
- 早期アクセスのサポート - 一部の变換は、さらなるテストと改良が行われるにつれて、早期アクセスとしてマークされる場合があります。

使用可能な AWS マネージド変換

次のカタログでは、現在利用可能な AWS マネージド変換をユースケース別にグループ化して一覧表示します。各テーブルには、変換名、適用する言語またはスタック、およびそのステータスが表示されます。変換をすばやく検索するには、このページの上部にある検索ボックスを使用して、名前、言語、またはユースケースで検索します。

このカタログは、AWSマネージド変換の正規リストです。Kiro Power、エージェントプラグイン、VS Code プラグイン、および CLI はすべて同じ変換を実行します。レジストリで利用できる変換を一覧表示するには、を実行します `atx custom def list`。

Note

「早期アクセス」とマークされた変換は機能しますが、お客様からのフィードバックに基づいて頻繁に更新される可能性があります。

ランタイムアップグレード

言語またはランタイムを新しいバージョンにアップグレードするか、ランタイムディストリビューションを変更します。

変換	言語/スタック	ステータス	説明
AWS/java-version-upgrade	Java	一般提供	Jakarta EE 移行、データベースドライバ、ORM フレームワーク、Spring エコシステムの更新などの依存関係のモダナイゼーションを使用して、任意のソース JDK バージョンから任意のターゲット JDK バージョンにビルドシステムを使用して Java アプリケーションをアップグレードします。チャットまたは <code>additionalPlanContext</code> パラメータを使用して、ターゲット JDK

変換	言語/スタック	ステータス	説明
			バージョンを指定します。
AWS/python-version-upgrade	Python	一般提供	機能とパフォーマンスを維持しながら、Python プロジェクトを Python 3.8 または 3.9 から Python 3.11、3.12、または 3.13 に移行します。チャットまたは additionalPlanContext パラメータを使用して、ターゲット Python バージョンを指定します。
AWS/nodejs-version-upgrade	Node.js	一般提供	Node.js アプリケーションを任意のソース Node.js バージョンから任意のターゲット Node.js バージョンにアップグレードします。チャットまたは additionalPlanContext パラメータを使用してターゲットバージョンを指定します。

変換	言語/スタック	ステータス	説明
AWS/lambda-nodejs-runtime-upgrade	Node.js (Lambda)	早期アクセス	AWS Lambda 関数を古い Node.js ランタイム (nodejs4.3 から nodejs22.x) から nodejs24.x にアップグレードし、Lambda ランタイムインターフェイスクライアント (RIC) と Node.js 24 言語ランタイムの重大な変更に対処します。
AWS/oracle-java-to-corretto	Java (JDK ディストリビューション)	早期アクセス	Oracle JDK から Amazon Corretto に Java プロジェクトを移行します。Oracle 固有の内部 APIs 標準の Java 同等のものに置き換え、ビルド設定 (Maven および Gradle) とコンテナベースイメージを更新し、商用 JVM フラグを削除して、ライセンスレポートを生成します。

変換	言語/スタック	ステータス	説明
AWS/ruby-upgrade	Ruby	早期アクセス	Ruby アプリケーションを Ruby 2.x から Ruby 4.0 にアップグレードし、そのフレームワークを Rails 8.0 または Sinatra 4.1 にアップグレードします。この変換は、Rails および ActiveRecord アプリ、Sinatra アプリ、スタンドアロン Gem で使用できます。変換は、各 Ruby バージョンバンプと一致するフレームワークのアップグレードをペアにして、テストスイートを実行します。その後、次のバージョンに進む前に障害を分離します。

SDK の移行

AWS SDK のメジャーバージョン間で移行します。

変換	言語/スタック	ステータス	説明
AWS/java-aws-sdk-v1-to-v2	Java	一般提供	Maven または Gradle を使用して、Java プロジェクト用の AWS SDK を v1 から v2 に

変換	言語/スタック	ステータス	説明
			アップグレードします。
AWS/python-boto2-to-boto3	Python	一般提供	公式の移行ドキュメントに基づいて、Python アプリケーションを boto2 から boto3 AWS に移行します。
AWS/nodejs-aws-sdk-v2-to-v3	Node.js	一般提供	Node.js バージョンを変更せずに、モジュールアーキテクチャ、ファーストクラス TypeScript サポート、パフォーマンスの向上のために、Node.js アプリケーションを AWS SDK for JavaScript v2 から v3 にアップグレードします。

フレームワークのアップグレードと移行

フレームワークを新しいバージョンにアップグレードするか、別のフレームワークに移行します。

変換	言語/スタック	ステータス	説明
AWS/spring-boot-version-upgrade	Java / バネブート	一般提供	古い Spring Boot アプリケーションを、ビルドファイルとスターター、プロパティ、Jackson 3 パッケージ、Spring Security 7 DSL、テ

変換	言語/スタック	ステータス	説明
			ストインフラストラクチャ、オブザーバビリティの依存関係など、ターゲット Spring Boot バージョンにアップグレードします。
AWS/JBoss-to-Spring-Boot	Java (JBoss または WildFly から Spring Boot)	早期アクセス	JBoss EAP または WildFly で実行されている Java EE または Jakarta EE エンタープライズアプリケーションを Spring Boot に移行し、クラウドネイティブでコンテナ化されたデプロイモデルのアプリケーションサーバーの依存関係を排除します。
AWS/early-access-angular-to-react-migration	対応角度	早期アクセス	Angular アプリケーションを React に変換します。
AWS/angular-version-upgrade	角度	早期アクセス	古い Angular アプリケーションを、コンポーネント、サービス、テンプレート、ルーティングなどのターゲット Angular バージョンにアップグレードします。

変換	言語/スタック	ステータス	説明
AWS/vue.js-version-upgrade	Vue.js	早期アクセス	Vue.js 2 から Vue.js 3 へのメジャーバージョンアップグレードを実行し、コンポーネント、状態管理、ルーティング、グローバル APIs。マイナー更新とパッチ更新は範囲外です。

GenAI とモデルの移行

生成 AI ワークロードを に移行します。

変換	言語/スタック	ステータス	説明
AWS/GenAI-to-Bedrock-Migration-Assessment	生成 AI ワークロード	早期アクセス	生成 AI ワークロードを評価して、サードパーティープロバイダー (OpenAI、Google Gemini、Anthropic direct、オープンソースモデル) から に移行します。SDK の使用状況とモデルを検出し、要件を明確にし、モデルマッピングを設計し、コストとリスクを見積もり、評価アーティファクトを生成します。評価のみ。ソースコードは変更されません。ま

変換	言語/スタック	ステータス	説明
			た、CrewAI、Lang Graph、Strands などのエージェントフレームワークについても説明します。

Language-to-language移行

あるプログラミング言語から別のプログラミング言語にコードベースを翻訳します。

変換	言語/スタック	ステータス	説明
AWS/vba-to-python-migration	VBA から Python	早期アクセス	openpyxl、pandas、tkinter UserForms、埋め込みロジックを同等の Python スクリプトとモジュールに移行します。PyQt6 ターゲット言語は Python 3.8 以降です。.bas、.cls、.frm、または .xlsm 埋め込み VBA コードを変換するときに使用します。

データベース移行

現在、データベース固有の AWS マネージド変換は利用できません。Microsoft SQL Server データベースおよび関連する .NET アプリケーションを Amazon Aurora PostgreSQL にモダナイズするには、「」を参照してください[the section called “SQL Server のモダナイゼーション”](#)。

オブザーバビリティ

ログ記録とモニタリングをネイティブサービスにモダナイズ AWSします。

変換	言語/スタック	ステータス	説明
AWS/early-access-log4j-to-slf4j-migration	Java (ログ記録)	早期アクセス	Logback バックエンドを使用して、Java アプリケーションを Log4j (1.x または 2.x) から SLF4J に移行します。ソースコード、依存関係管理 (Maven と Gradle)、およびログ設定ファイルを処理し、コンパイル、テスト、残差インポートスキャンによって検証します。
AWS/datadog-monitors-to-cloudwatch-alarms	コードとしてのインフラストラクチャ/モニタリング	早期アクセス	AWS サービスメトリクスとカスタムメトリクスを追跡する DataDog メトリクスモニターを、Infrastructure as Code としてネイティブ CloudWatch アラームに移行します。CloudFormation YAML、CDK TypeScript、Terraform HCL を生成します。

アーキテクチャ

アプリケーションを再設計、再プラットフォーム、または最適化します AWS。

変換	言語/スタック	ステータス	説明
AWS/java-performance-optimization	Java (パフォーマンス)	一般提供	JFR プロファイリングデータを分析して CPU とメモリのホットスポットとアンチパターンを検出し、ターゲットを絞ったコード修正を適用することで、Java アプリケーションのパフォーマンスを最適化します。JFR データを収集する手順については、 JFR ランタイムガイド を参照してください。
AWS/early-access-java-x86-to-graviton	Java (Arm64/Graviton)	早期アクセス	Graviton プロセッサ用の Arm64 アーキテクチャとの Java AWS アプリケーションの互換性を検証し、依存関係の更新、アーキテクチャ固有のコードパターンの検出、ソースコードが利用可能になったときのネイティブライブラリの再コンパイルにより、非互換性を解決します。最新の Java アプリケーショ

変換	言語/スタック	ステータス	説明
			ンの多くは、既に Arm64-compatible があります。
AWS/oracle-service-bus-to-aws	Java / サーバーレスとの統合	早期アクセス	Oracle Service Bus (OSB) と BPEL プロセス設定を AWS ネイティブサーバーレスアーキテクチャに移行し、Amazon API Gateway、AWS Lambda、AWS Step Functions を使用してデプロイ可能な CDK TypeScript プロジェクトを生成します。

変換	言語/スタック	ステータス	説明
AWS/mulesoft-to-aws-native	Java/MuleSoft からサーバーレスへ	早期アクセス	MuleSoft Mule 3.x および 4.x アプリケーションを AWS サーバーレスアーキテクチャ (Amazon API Gateway、AWS Lambda、AWS Step Functions) に変換し、SnapStart を使用して AWS Lambda 上の Java 17 をターゲットにします。変換は、フロートリガーを AWS イベントソースにマッピングし、DataWeave 式と MEL (MuleSoft 式言語) 式を AWS 同等のものに変換し、コネクタを AWS SDK for Java v2 呼び出しに置き換えます。SAM テンプレート、AWS Lambda ハンドラー、JUnit 5 テストを生成します。

変換	言語/スタック	ステータス	説明
AWS/payshield-hsm-to-aws-payment-cryptography	Java (HSM から AWS Payment Cryptography)	一般提供	Java アプリケーションを Thales PayShield ハードウェアセキュリティモジュール (HSM) プロトコルから AWS Payment Cryptography SDK v2 に移行します。移行には、コマンドマッピング、キー移行、依存関係のセットアップ、パリティテストが含まれます。

コードベース分析

コードベースとポートフォリオを分析してモダナイゼーションを計画します。これらの変換はレポートを生成し、コードを変更しません。

変換	言語/スタック	ステータス	説明
AWS/comprehensive-codebase-analysis	複数言語	一般提供	コードベースの詳細な静的分析を実行して、動作分析、アーキテクチャドキュメント、ビジネスインテリジェンスを技術的負債インサイトに重点を置いて組み合わせた、階層的で相互参照型のドキュメントを生成します。

変換	言語/スタック	ステータス	説明
AWS/agentic-readiness-analysis	複数言語	早期アクセス	APIs、アイデンティティ、状態管理、ヒューman-in-the-loop、オブザーバビリティなど、システムが AI エージェントによって安全に呼び出される準備ができているかどうかを評価します。
AWS/modernization-readiness-analysis	複数言語	早期アクセス	クラウドネイティブの成熟度ギャップのポートフォリオをスキャンし、結果を AWS モダナイゼーションパスにマッピングします。
AWS/portfolio-agentic-readiness-analysis	ポートフォリオ	早期アクセス	アプリケーションのポートフォリオ全体で個々のエージェント準備状況分析レポートを集約し、クロスカットブロッカー、共有修復パターン、優先順位付けされたレコメンデーションを特定します。

変換	言語/スタック	ステータス	説明
AWS/portfolio-modernization-readiness-analysis	ポートフォリオ	早期アクセス	ポートフォリオ全体の個々のモダナイゼーション分析レポートを、優先順位付けされた移行ウェーブと推奨されるモダナイゼーション経路を含む統合ロードマップに集約します。

変換	言語/スタック	ステータス	説明
AWS/business-rules-extraction	複数言語	早期アクセス	この変換では、モノリシックコードベースを静的に分析して、ビルド、実行、またはソースコードの変更を必要とせずに、書き換え可能なドキュメントを作成します。システムを境界ドメインに分解し、ドメインごとのビジネスルール、ワークフロー、クロスカットの懸念、データベース構造を抽出します。出力には、モダナイゼーションまたは完全な書き換えをガイドするための、マシン読み取り可能なマニフェスト、インタラクティブなダッシュボード、ドメインごとの言語に依存しない実装仕様が含まれます。

AWS マネージド変換のカスタマイズ

`additionalPlanContext` 設定パラメータを使用して追加のコンテキストを指定することで、組織の特定のニーズを満たすように AWS マネージド変換をカスタマイズできます。

例: Java バージョンアップグレードのカスタマイズ

```
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.
```

例: AWS SDK 移行のカスタマイズ

```
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.
```

一般的なユースケース

このセクションでは、AWS Transform step-by-stepのガイダンスを提供します。

Lambda ランタイムのアップグレード

Lambda は古い言語ランタイムを定期的に廃止するため、チームは Lambda 関数をサポートされているバージョンにアップグレードする必要があります。AWS Transform custom は AWS、マネージド言語バージョンアップグレード変換を使用してこれらのアップグレードを自動化できます。たとえば、Python 3.9 で記述された Lambda 関数を Python 3.13 にアップグレードするには、AWS/python-version-upgrade変換を使用します。Java (AWS/java-version-upgrade) と Node.js () でも同様の変換が利用できAWS/nodejs-version-upgrade、最も一般的な Lambda ランタイム言語をカバーしています。

単一の Lambda 関数をインタラクティブにアップグレードするには、関数のソースリポジトリのルートに移動し、で AWS Transform CLI を起動しますatx。インタラクティブセッションで、必要なアップグレードについて説明します。たとえば、「この Python 3.9 Lambda 関数を Python 3.13 にアップグレードする」などです。エージェントは、適切な AWSマネージド変換の選択、コードの分析、言語構文の更新、依存関係の調整、廃止された API コールのモダナイズを行います。pytest やなどのビルドまたは検証コマンドを指定python -m py_compile *.pyして、エージェントが変換されたコードのコンパイルを検証し、テストに合格できるようにします。インタラクティブセッ

ションを通じて、最終的な結果を承諾する前に、エージェントの提案された変更を確認し、フィードバックを提供し、調整をリクエストできます。

多くのリポジトリで Lambda ランタイムをアップグレードする必要がある場合は、非インタラクティブモードを使用して大規模な変換を実行します。リポジトリごとに、ユーザー入力なしで実行する適切なフラグを使用して CLI を呼び出します。たとえば、Node.js 16 Lambda 関数を Node.js 22 にアップグレードするには:

```
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 フラグは変換を非インタラクティブに実行し、すべてのツール実行を -t 信頼するため、プロンプトは必要ありません。シンプルなシェルループを使用して、数十または数百のリポジトリでこのコマンドをスクリプト化できます。各変換が完了したら、Git 差分を確認して変更を確認し、テストスイートを実行してアップグレードされた関数が正しく動作することを確認します。AWS Transform custom を CI/CD パイプラインに統合して、非推奨の Lambda 関数を継続的に特定することもできます。このアプローチは、廃止されたランタイムの即時の問題に対処するだけでなく、end-of-support ステータスに達する前に関数を検出してアップグレードする継続的なプロセスを確立します。

中央プラットフォームチームは、[AWS 変換ウェブアプリケーション](#)を通じてキャンペーンを作成して、変換を定義し、ターゲット Lambda リポジトリを指定し、組織全体の進捗状況を追跡できます。AWS Batch を使用して本稼働グレードのスケールドデプロイソリューションを構築する詳細なチュートリアルについては、[AWS 「Transform custom を使用したスケーラブルなコードモダライゼーションソリューション」の構築](#)を参照してください。

AWS Transform カスタムエンドポイントとインターフェイスエンドポイント (AWS PrivateLink)

インターフェイス VPC エンドポイントを作成することで、VPC と AWS Transform カスタムの間にプライベート接続を確立できます。インターフェイスエンドポイントは、インターネットゲートウェイ [AWS PrivateLink](#)、NAT デバイス、VPN 接続、または Direct Connect 接続なしで AWS Transform カスタムサービスにプライベートにアクセスできるテクノロジーである [AWS PrivateLink](#) を利用しています。VPC と AWS Transform カスタム間のトラフィックは Amazon ネットワークを離れません。

各インターフェイスエンドポイントは、サブネット内の 1 つ、または複数の [Elastic Network Interface](#) によって表されます。

詳細については、「Amazon VPC ユーザーガイド」の「[インターフェイス VPC エンドポイント \(AWS PrivateLink\)](#)」を参照してください。

Note

- AWS Transform カスタムは S3 API コールを行うため、Amazon S3 AWS PrivateLink の統合を有効にする必要があります。詳細な手順については、[AWS PrivateLink Amazon S3 のドキュメント](#)」を参照してください。AWS Transform カスタムの使用中に S3 アクセスの問題が発生した場合は、[トラブルシューティングガイド](#)を参照してください。
- AWS PrivateLink プライベート DNS 機能を使用していない場合 (「[プライベート DNS](#)」を参照)、以下を行う必要があります。
 - VPC インターフェイスエンドポイントへのルーティングを設定する (「[VPC インターフェイスエンドポイントへのルーティング](#)」ドキュメントを参照)
 - ATX_CUSTOM_ENDPOINT 環境変数を設定して、カスタムドメインを指定します。次に例を示します。

```
ATX_CUSTOM_ENDPOINT=https://transform-custom.<region>.api.aws atx
```

AWS Transform カスタム VPC エンドポイントに関する考慮事項

AWS Transform カスタムのインターフェイス VPC エンドポイントを設定する前に、Amazon VPC ユーザーガイドの[インターフェイスエンドポイントのプロパティと制限](#)を確認してください。

AWS Transform カスタムは、インターフェイスエンドポイントを介したすべての API アクションの呼び出しをサポートしています。

前提条件

以下の手順を開始する前に、次の項目が揃っていることを確認してください。

- リソースを作成および設定するための適切なアクセス許可を持つ AWS アカウント。
- AWS アカウントに既に作成された VPC。
- AWS サービス、特に Amazon VPC と AWS Transform カスタムに精通していること。

AWS Transform カスタムのインターフェイス VPC エンドポイントの作成

Amazon VPC コンソールまたは AWS Command Line Interface () を使用して、AWS Transform カスタムサービスの VPC エンドポイントを作成できます AWS CLI。詳細については、「Amazon VPC ユーザーガイド」の [インターフェイスエンドポイントの作成](#) を参照してください。

このサービス名を使用して、AWS Transform カスタム用に次の VPC エンドポイントを作成します。

- `com.amazonaws.region.transform-custom`

region を、`com.amazonaws.us-east-1.transform-custom` などの AWS Transform カスタム CLI を使用する AWS リージョンに置き換えます。

詳細については、「Amazon VPC ユーザーガイド」の「[インターフェイスエンドポイントを介したサービスへのアクセス](#)」を参照してください。

AWS Transform カスタムの VPC エンドポイントポリシーの作成

AWS Transform カスタムへのアクセスを制御するエンドポイントポリシーを VPC エンドポイントにアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- アクションを実行できるリソース。

詳細については、「Amazon VPC ユーザーガイド」の「[VPC エンドポイントでサービスへのアクセスを制御する](#)」を参照してください。

例: AWS Transform カスタムアクションの VPC エンドポイントポリシー

AWS Transform カスタムのエンドポイントポリシーの例を次に示します。エンドポイントにアタッチすると、このポリシーは、すべてのリソースのすべてのプリンシパルに対して、リストされた AWS Transform カスタムアクションへのアクセスを許可します。

```
{
  "Statement": [
    {
      "Principal": "*",
```

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

オンプレミスコンピュータを使用して AWS Transform カスタムエンドポイントに接続する

このセクションでは、オンプレミスコンピュータを使用して AWS VPC の AWS PrivateLink エンドポイントを介して AWS Transform カスタム に接続するプロセスについて説明します。

1. [オンプレミスデバイスと VPC の間に VPN 接続を作成します。](#)
2. [AWS Transform カスタムのインターフェイス VPC エンドポイントを作成します。](#)
3. [インバウンド Amazon Route 53 エンドポイントをセットアップします。](#) これにより、オンプレミスデバイスから AWS Transform カスタムエンドポイントの DNS 名を使用できます。

トラブルシューティング

このセクションでは、AWS 変換カスタムの一般的な問題をトラブルシューティングするためのガイドランスを提供します。

ログの場所

AWS Transform CLI は、次の場所にログを保持します。

会話ログ:

Linux and macOS

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

Windows

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

これらのログには、特定の変換実行をデバッグするための会話履歴がすべて含まれています。

サブエージェントログ:

Linux and macOS

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

Windows

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

これらのログには、変換中にメインエージェントがスポンするサブエージェントからの出力が含まれています。サブエージェントを直接管理する必要はありません。

開発者デバッグログ:

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
```

これらのログは、CLI オペレーションとエラーに関する詳細情報を提供します。各ログファイルには、ロールオーバー前に 5MB の制限があります。このディレクトリには、debug1.log や debug2.log など、複数のデバッグログが存在する場合があります。

一般的な問題

インストールの問題:

インストールが失敗した場合は、Node.js 22 以降がインストールされていることを確認します。

```
node --version
```


必要に応じて、<https://nodejs.org/en/download> から Node.js をダウンロードします。

認証の問題:

AWS 認証情報が正しく設定されていることを確認します。

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

IAM ユーザーまたはロールに必要な `transform-custom:*` アクセス許可があることを確認します。

ネットワーク接続の問題:

接続エラーが発生した場合は、必要なエンドポイントへのネットワークアクセスを確認します。

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

インターネットが制限された環境で作業する場合は、ファイアウォールルールを更新してこれらの URLs。

リージョン設定の問題:

リージョン関連のエラーが発生した場合:

- リージョンがサポートされていることを確認します。
- 設定を上書きする可能性のある環境変数を確認します。 `echo $AWS_REGION`
`$AWS_DEFAULT_REGION`
- リージョン設定を確認します。 `aws configure get region`
- 必要に応じてリージョンを更新します。 `aws configure set region <your-region>`
- デバッグログでリージョン解決の詳細を確認する

Git の問題:

Git がインストールされ、リポジトリが git ソース管理下にあることを確認します。

```
git --version
```

```
git status
```

AWS 変換カスタムでは、リポジトリが git ソース管理対象である必要があります。

変換実行の問題:

変換が失敗した場合:

1. 会話ログを確認する `~/.aws/atx/custom/<conversation_id>/logs/`
2. 変換出力でビルドまたはテストの失敗を確認する
3. ビルドコマンドがプロジェクトに対して正しいことを確認する
4. インタラクティブモードで変換を実行してフィードバックを提供してみてください

会話の再開に関する問題:

会話を再開できない場合:

- 会話が 30 日未満であることを確認する
- 会話 ID が正しいことを確認する
- ネットワーク接続があることを確認する

サポート情報

詳細については、[AWS コンソール](#)から AWS サポートを参照してください。

サポートチケットを開くときは、以下を含めます。

- からの会話ログ `~/.aws/atx/custom/<conversation_id>/logs/`
- からのサブエージェントログ `~/.aws/atx/custom/<conversation_id>/logs/subagents/`
- からのログのデバッグ `~/.aws/atx/logs/`
- 問題を再現するステップ
- AWS 変換 CLI バージョン (`atx --version`)

S3 アクセスの問題

AWS Transform カスタムサービスは S3 の署名付き URLs を提供し、クライアントマシンとの間で大きな変換ファイルのアップロードとダウンロードを容易にします。つまり、CLI は AWS

Transform サービスエンドポイントとやり取りするだけでなく、S3 サービスエンドポイントともやり取りします。これらのやり取りでは署名付き URLs を使用するため、クライアントの IAM 認証情報に S3 アクセス許可は必要ありませんが、ローカルマシンのプロキシサーバー設定やネットワークの S3 VPC エンドポイントポリシーによってこのトラフィックが制限されることがあります。

変換ファイルのダウンロードとアップロードに関連するツールの障害や S3 アクセス拒否の根本原因となる詳細なエラーメッセージ[the section called “ログの場所”](#)については、「」を参照してください。

https_proxy 環境変数と no_proxy環境変数を活用する VPN/プロキシサーバーを使用している場合は、no_proxy環境変数値に次の値を追加して S3 のプロキシをバイパスし、カスタムサービスエンドポイントを AWS 変換することを検討してください。次に例を示します。

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"
```

VPC で [AWS Amazon S3 用の PrivateLink](#) を使用している場合、S3 トラフィックを制限するようにポリシーが定義されている可能性があります。S3 VPC エンドポイントポリシーで AWS、変換カスタムサービス所有バケットの GetObjectおよび PutObjectオペレーションが許可されていることを確認してください。これは、VPC エンドポイントポリシーに次のステートメントを追加することで実現できます。

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

ポリシーの明示的なDenyステートメントは、Allowステートメントよりも優先されます。アクセスを制限している可能性があるDenyステートメントについては、S3 VPC エンドポイントポリシーを確認してください。

AWS CLI バージョン履歴の変換

AWS Transform CLI の現在および過去のリリースの詳細については、次の表を参照してください。この表には、各バージョンのバージョン、リリース日、リリースノートが含まれています。

バージョン	リリース日	リリースノート
3.6.0 (最新)	2026 年 7 月 17 日	• <code>atx custom def learnings</code> サブコマンドを追加しました。これにより、インタラクティブ TUI セッションが開き、以前の実行に基づいて変換定義に対して自動的に生成された学習を表示および管理できます。 • 変換の実行中に使用状況テレメトリコレクションを追加しました。テレメトリを無効にするには、<code>ATX_DISABLE_TELEMETRY=true</code> 環境変数を設定します。詳細については、「使用状況テレメトリ」を参照してください。 • (~ホームディレクトリ) パスを使用したファイルオペレーションが予期しないディレクトリに解決される Windows の問題を修正しました。これにより、セッション中にファイルの読み取り/書き込みが失敗しました。 • 特殊文字を含む MCP サーバーとツール名によって接続が失敗する問題を修正しました。CLI は、有効な文字のみを使用するように名前を自動的にサニタイズするようになりました。
3.5.0	2026 年 7 月 8 日	• Windows のネイティブサポートが追加されました。Windows で AWS Transform Custom コマンドを直接実行できるようになりました。(atx ct コマンドは Windows ではまだサポートされていません) • エージェントの準備状況とモダナイゼーションの準備状況の分析に好み が反映され atx ct analysis run するように、ステアリング入力を に追加しました。を戦略的フレーミング--context に使用し、エージェント対応のためのエージェントアクセス--agent-scope を設定し、--prefer/--avoid を使用してモダナイゼーションのレコメンデーションを特定のテクノロジーに対して、または特定のテクノロジーから遠ざけます。

バージョン	リリース日	リリースノート
		• スキル形式の変換定義にscripts/フォルダを含めることができ、SKILL.mdおよび リファレンスと一緒にヘルパー スクリプトをバンドルできるようになりました。 • 修復が完了するまで待機atx ct remediation createする --waitを に追加しました。 • ワンショット実行のヘッドレスモードを追加しました。を実行してプロンプトatx -x "prompt" -tを渡し、レスポンスを取得し、インタラクティブプロンプトなしで終了します。ヘッドレスモードでは、すべてのツールを信頼する必要があります。
3.4.1	2026 年 7 月 2 日	• バージョン 3.4.0 で導入されたインタラクティブセッションの問題を修正しました。エージェントが応答を完了すると、CLI は入力プロンプトに戻る代わりに、「思考」プロンプト (UI に省略記号で表示) に留まる場合があります。CLI は、各レスポンスの後に入力プロンプトに戻るようになりました。

バージョン	リリース日	リリースノート
3.4.0 (非推奨)	2026 年 6 月 29 日	⚠ [既知の問題] インタラクティブセッションでは、エージェントが応答を終了した後、CLI は「Thinking」プロンプト (UI に省略記号で表示) にとどまることがあります。これにより、次のメッセージを入力できなくなります。この問題を解決するには、 を実行してバージョン 3.4.1 以降にアップグレードしますatx update。 - スクリプトと自動化のための完全なatx ctコマンドサーフェスの機械読み取り可能なマニフェストatx ct schemaである を追加しました。 - ソースを追加または更新するときにトークンスコープの検証が追加されました。分析中に後で失敗するのではなく、認証情報やアクセス許可の問題がすぐに表示されます。 - 自動修復プルリクエストが改善され、結果コンテキストが含まれ、よりレビュー可能な PRs。 - 現在のアクティビティ、エージェントの推定分数、およびサブエージェントのアクティビティに関するリアルタイムコンテキストによる CLI 進行状況のフィードバックが改善されました。 - 環境ATX_GIT_COMMITTER_EMAIL 変数 ATX_GIT_COMMITTER_NAME および を使用して、設定可能なチェックポイントコミット作成者 ID を追加しました。 - さまざまな小さなバグ修正と改善。
3.3.0	2026 年 6 月 18 日	- atx ct コマンド間で整合性--jsonと--helpサポートが追加されました。 - セキュリティエージェントのセットアップを修正し、セットアップ中にエージェントスペースが作成され、ティアダウン時に削除されます。 - さまざまな小さなバグ修正と改善。

バージョン	リリース日	リリースノート
3.2.0	2026 年 6 月 15 日	- すべてのユーザーの標準 CLI ディストリビューションに継続的なモダナイゼーションコマンド (atx ct) を追加しました。 - サーバーレスポンスを待っている間に「Thinking...」スピナーを表示するようにサブエージェントの実行を改善し、作業中であることを示す明確な視覚的なフィードバックを提供しました。 - エージェント出力の 3 秒後に「作業中...」で自動的に再起動するようにスピナーが改善されたため、処理ステップ間でターミナルがフリーズすることはありません。 - 微妙な光沢アニメーションで思考スピナーを改善し、より洗練された外観を実現しました。
3.1.0	2026 年 6 月 3 日	- 最大検証ループを設定する ATX_MAX_VALIDATION_LOOPS 環境変数を追加しました。 - インストールされたバージョンがすでにレジストリバージョンより新しい場合、更新チェックで誤った「利用可能な更新」警告が表示されなくなりました。 - 修正され、非センシカルな確認プロンプトが表示され delete --transformation-name "" なくなりました。空の値は、明確な検証エラーで拒否されるようになりました。 - 認識されないサブコマンドが、混乱する内部エラーメッセージの代わりに「不明なコマンド」を表示するようになりました。 - 更新チェックに文字列の不等式の代わりに数値比較を使用するようにバージョン比較を修正しました。

バージョン	リリース日	リリースノート
3.0.0	2026 年 5 月 21 日	- 名前が見つからないときにレジストリから同様の名前を提案するプリフライト変換名検証が追加され、別のリストコマンドを必要とせずにタイプミスをキャッチできるようになりました。 - シェルコマンドの動作と理由を示す、オプションの説明を追加しました。 - 信頼設定バナーが改善され、ツールのアクセス許可に関する詳細情報が追加されました。 - do-not-learn フラグを使用するときに learning-disabled バナーを追加し、学習機能とその有効化方法を可視化しました。 - 重大な変更: Node.js 22 以降が必要になりました。古いバージョンを使用している場合は、https://nodejs.org/ からアップグレードします。

バージョン	リリース日	リリースノート
2.0.0	2026 年 5 月 14 日	- リモート MCP サーバーのサポートを追加 — 自動 SSE フォールバック、ヘッダーの環境変数拡張によるベアラー トークン認証、プロキシサポート (HTTP_PROXY/HTTPS_PROXY) を使用して HTTP および HTTPS MCP サーバーに 接続します。 - のエイリアスとしてatx resumeサブコマンドを追加しま したatx --resume。 - 出力制御によるマルチファイルテキスト検索用のgrepツ ールを追加 — 正規表現パターン、ファイル glob フィルタリ ング、大文字と小文字を区別しない検索、設定可能な深度/ 結果制限、出力切り捨てをサポートしています。 - MCP の実装が改善され、標準の MCP SDK を使用して、 より一貫した動作を実現し、将来の MCP 機能を簡単にサ ポートできるようになりました。 - シェル出力処理の向上 — 出力は、ハードエラーを引き起こ すのではなく、ストリームあたり 30K 文字で適切に切り捨 てられ、コンテキストの出力の開始と終了が保持されるよ うになりました。 - インタラクティブセッション中の Ctrl+C が、現在のオペ レーションを中止してプロンプトに戻る代わりに、Linux/ macOS で CLI を途中で終了する問題を修正しました。 - コマンドに認可が必要な理由とは別に、信頼が利用できな い理由を示す信頼拒否メッセージを修正しました。 - コマンドが にあるときに、危険なコマンド警告が信頼につ いて言及しなくなりましたalwaysPromptCommands 。

バージョン	リリース日	リリースノート
1.10.0	2026 年 4 月 29 日	- クロスツールガイダンスによるツールの説明が改善され、エージェントが各タスクに最適なツールを選択し、不要なシェル使用量を削減できるようになりました。 <code>alwaysPromptCommands</code> で追加 <code>trust-settings.yaml</code> — 信頼設定に関係なく、常に明示的なアクセス許可を必要とするシェルコマンドパターンを設定します。複合式の各サブコマンドに対する glob ワイルドカードとマッチングをサポートします。 <code>trustedShellCommands</code> はこれらのパターンを上書きできます。自動パイプラインの中断を避けるため、非インタラクティブモードでは強制されません。 - 相対パスが絶対パスに解決されない問題を修正しました。
1.9.0	2026 年 4 月 23 日	ツールの承認プロンプトが改善されました。各オプションでは、インラインでの動作が表示され、承認が必要な理由を説明する「理由」メッセージが表示され、信頼の確認には永続的な信頼 <code>trust-settings.yaml</code> のための に関するヒントが含まれています。
1.8.0	2026 年 4 月 16 日	- ユーザー入力とエージェント出力の視覚的な区別を明確に ATX: するため、エージェント出力プレフィックスを > から に変更しました。 - CLI の再試行が枯渇した場合の再開コマンドの表示を修正しました。 - 別の作業ディレクトリからの再開を修正しました。インタラクティブモードでは、 は修正されたパスをプロンプトします。非インタラクティブモードでは、 は説明的なエラーで高速に失敗します。

バージョン	リリース日	リリースノート
1.7.1	2026 年 4 月 9 日	- 認証情報プロバイダーリージョンがプロファイルの と異なる場合の SSO ログインエラー (「セッショントークンが見つからないか無効」) を修正しましたsso_region。 - 認証情報のキャッシュを修正し、ディスクで認証情報をローテーションしたときに長時間実行されるセッションが失敗しなくなりました。
1.7.0	2026 年 4 月 2 日	- 入力検証エラー時にコード 1 で終了するexecコマンドを修正し、スクリプトと CI/CD パイプラインで CLI の信頼性を高めました。 - サービスからの警告メッセージの表示を追加しました。 - AWS 認証情報プロファイルの MFA プロンプトサポートを追加しました。 - HTTP(S) プロキシ設定を尊重する STS 認証情報呼び出しを修正しました。
1.6.0	2026 年 3 月 18 日	- exec および interactive コマンドのエージェント分予算を設定する--limit <minutes> オプションを追加しました。制限に達すると、CLI は使用状況を表示し、再開手順で終了します。 - 会話ログの最初のユーザーメッセージのログ記録を修正しました。
1.5.0	2026 年 3 月 11 日	- 同時タスク実行のサポートが追加され、変換が高速化されました。 - 各会話の最後に Agent Minutes 使用状況の表示が追加され、いつでも使用状況をチェックするインタラクティブモードで/usageのコマンドが追加されました。

バージョン	リリース日	リリースノート
1.4.0	2026 年 3 月 2 日	- サポートされていないファイルで変換定義を保存するときのエラーメッセージが改善されました。 - 接続が中断されたときに CLI がハングしないように、サービス呼び出しにタイムアウトを追加しました。 - 正常に完了した後の CLI プロセスのハングを修正しました。 - 会話を再開する際のリージョンの検証を修正しました。
1.3.0	2026 年 2 月 19 日	- MCP サーバーパスの制限を削除 — ユーザーは MCP 設定で commandプロパティの任意のパスを指定できるようになりました。 - MCP サーバー接続の安定性が向上しました。
1.2.0	2026 年 2 月 9 日	- 環境変数 (https_proxy 、 no_proxyなど) を介したプロキシ設定のサポートが追加されました。 - 一般的な安定性の向上とバグ修正。
1.1.1	2026 年 1 月 16 日	バグ修正および安定性の向上。
1.1.0	2026 年 1 月 13 日	CLI AWS リージョン設定のサポート、認証情報の早期検証チェック、および固定キャンペーン再開機能を追加しました。
1.0.1	2025 年 12 月 18 日	正常なシャットダウン、よりクリーンなエラーログ、一般的な改善。
1.0.0	2025 年 12 月 1 日	AWS Transform CLI の初回リリース。

AWS 継続的なモダナイゼーションを変革する

AWS Transform Continuous Modernization とは

AWS 継続的なモダナイゼーションを変換することで、ソースコードリポジトリの分析と修復が可能になります。GitHub 組織、GitLab グループ、Bitbucket ワークスペース、ローカルリポジトリを接続し、自動化された分析を実行して、コードベース全体の技術的負債、セキュリティの脆弱性、モダナイゼーションの機会、エージェントの準備状況を特定できます。

すべての分析と修復は、認証情報 AWS アカウント を使用して で実行されます。ソースコードはお客様の管理下に置かれます。

Note

Kiro Power またはエージェントプラグインから始めることをお勧めします。エージェントスキルは、インフラストラクチャのプロビジョニング、ソース設定、分析の実行、結果のトリガー、修復など、継続的なモダナイゼーションのセットアップ、オンボーディング、継続的な使用を調整します。インストール手順については、「[デベロッパーツール](#)」を参照してください。

主な機能

AWS 継続的なモダナイゼーションを変換すると、次の機能が提供されます。

- Tech Debt Analysis — 古い依存関係、セキュリティの脆弱性、コード品質の問題、モダナイゼーションの機会がないリポジトリをスキャンします。パッケージマニフェストまたは包括的なコードレベルの分析全体でクイックメタデータスキャンを実行します。環境に合わせた変換定義を使用して、カスタム分析基準を定義します。
- 自律修復 — 検証済みのプルリクエストを大規模に生成します。各検出結果は、関連する変換定義を使用して自動修正できます。修復はブランチを作成し、GitHub、GitLab、Bitbucket 全体で PRs/MRs を自動的に開きます。
- レポート — 重要度、リポジトリ、分析タイプ別に結果を示す HTML レポートを生成します。修復の進行状況と結果の解決を経時的に追跡します。

- 継続的モニタリング — Amazon EventBridge スケジューラを使用して定期的な分析をスケジュールします。毎日、毎週、またはカスタムの cron スケジュールを設定して、新しい問題がないかポートフォリオを継続的にモニタリングします。

分析タイプ

AWS 継続的なモダナイゼーションの変換では、モダナイゼーションのニーズに対応するために、次の分析タイプがサポートされています。

型	説明
rapid-techdebt-analysis	古いバージョンと古い依存関係を識別するための、パッケージマニフェスト (pom.xml、package.json、requirements.txt) のメタデータのみの高速スキャン。ソースコードを分析しません。
tech-debt-comprehensive	AWS Transform エージェントを使用した詳細なコードレベルの技術的負債分析。ソースコードを調べて、負債パターン、コード品質の問題、アーキテクチャの懸念、改善の機会を特定します。
security	セキュリティエージェントを使用した AWS セキュリティ脆弱性と CVE 検出。ソースコードと依存関係をスキャンして、既知の脆弱性、安全でないコーディングパターン、悪用可能な弱点がないかどうかを確認します。1 回限りのインフラストラクチャのセットアップが必要です。
agentic-readiness	AI とエージェントの統合の準備状況の評価。インフラストラクチャとプラットフォーム、アプリケーションアーキテクチャ、データ基盤、Identity/Security/Governance。

型	説明
modernization-readiness	クラウドモダナイゼーションの機会の評価。インフラストラクチャ、アプリケーション、データ、セキュリティ、運用の各ディメンションにわたる準備状況进行评估します。コンテナ化、サーバーレス移行、プラットフォームアップグレードの候補を特定します。
custom	任意の変換定義 (TD) を分析として実行します。このタイプを使用して、独自の分析基準を定義したり、組み込みタイプでカバーされていない AWS マネージド変換を実行したりできます。

重要な概念の理解

[Sources] (出典)

ソースは、リポジトリの場所を継続的なモダナイゼーションに伝えます。サポートされているソースタイプ:

- GitHub — 個人用アクセストークンを持つ組織 (クラシック)
- GitLab — セルフホストインスタンスを含むグループまたはユーザー
- Bitbucket — Workspaces (クラウド) またはプロジェクト (データセンター)
- Local — git リポジトリを含む親ディレクトリ

リポジトリ

リポジトリは、スキャンソースによって検出されます。検出後、ソース、ラベル、またはその他の基準でフィルタリングしたり、組織 (チーム、優先度、移行ウェーブ) にラベルを適用したり、分析や修復のために特定のリポジトリをターゲットにしたりできます。

分析

分析は、特定の分析タイプを使用する 1 つ以上のリポジトリのスキャンです。各分析では、コードの問題を識別する検出結果が生成されます。分析はオンデマンドで実行することも、自動的に実行するようにスケジューリングすることもできます。分析タイプには、rapid-techdebt-analysis、tech-debt-comprehensive、security、agentic-Readiness、モダナイゼーション-Readiness、Custom などがあります。各分析は、ステータス (保留中、実行中、完了、キャンセル、失敗) とスキャンしたリポジトリを追跡します。

検出結果

結果は分析の結果です。各検出結果には、重要度 (高、中、低)、ステータス (オープン、却下、または廃止)、およびそれを修復できる修正変換 (自動修正可能な場合) が含まれます。新しい検出結果はオープンとして始まります。ユーザーは、理由のある検出結果を却下できます。再分析では、解決された検出結果は自動的に廃止としてマークされます。

修復

修正では、変換定義を適用して検出結果を修正します。3 つのモード: 検出結果ベース (各検出結果は独自の修正変換を使用)、TD オーバーライド (指定された検出結果の変換定義を上書き)、および直接 TD (検出結果なしでリポジトリに対して変換を実行)。出力はソースプロバイダー (GitHub PR、GitLab MR、Bitbucket PR、またはローカルブランチ) によって異なります。

変換定義

変換定義には、特定のコード変換を実行するために必要な手順と知識が含まれています。継続的なモダナイゼーションでは、カスタム分析 (`--type custom --transformation-name name`) と修復 (`()`) に変換定義を使用します `--transformation-name name`。で使用可能な変換定義を一覧表示します `atx custom def list`。

AWS Transform の継続的なモダナイゼーションの仕組み

AWS トランスフォームの継続的なモダナイゼーションは通常、複数のコードベースで継続的な分析と修復が必要な大規模なプロジェクトで使用されます。チームは通常、このワークフローに従います。

1. Connect Sources — GitHub 組織、GitLab グループ、Bitbucket ワークスペース、またはローカルディレクトリをソースとして追加します。認証トークンに適切なスコープを提供します。

2. リポジトリの検出 — 検出スキャンを実行して、ソース内のすべてのリポジトリを列挙します。ラベルを使用して、チーム、優先度、または移行ウェーブ別にリポジトリを整理します。
3. 分析の実行 — ポートフォリオ全体で分析を実行します。迅速な結果を得るにはクイックスキャンを選択し、詳細な結果を得るには包括的な分析を選択します。セキュリティ分析には、1 回限りのインフラストラクチャ設定が必要です。
4. 結果のトリアージ — 重要度、リポジトリ、分析タイプ別に結果を確認します。文書化された理由で誤検出を却下します。分析を再実行して、解決された問題を古いものとしてマークします。
5. 修復 — 検出結果を修正するための修復を作成します。継続的なモダナイゼーションにより、プル/マージリクエストが開きます。修復ステータスを追跡し、失敗を再試行します。
6. 継続的分析の設定 — Amazon EventBridge スケジューラを使用して定期的な分析をスケジュールします。分析頻度 (毎日、毎週、またはカスタム cron) を設定して、ポートフォリオに新しい問題がないか継続的にモニタリングします。自動修復と組み合わせて、時間の経過とともにコードの状態を維持します。

コンピューティングオプション

継続的なモダナイゼーションは、分析と修復を実行するための 3 つのコンピューティングオプションをサポートしています。Kiro Power and Agent プラグインのエージェントスキルは、各オプションにインフラストラクチャを設定するのに役立ちます。

Note

コンピューティングオプションに関係なく、すべての分析と修復は認証情報 AWS アカウントを使用して行われます。ソースコードはお客様の管理下に置かれます。

Local (デフォルト)

デフォルトでは、分析はローカルマシンで実行されます。このオプションでは、追加のインフラストラクチャは必要ありません。サーバーはローカルで実行され、ローカルコンピューティングリソースを使用して分析を実行します。ツール、小さなリポジトリ、または個々の使用を試すのに適しています。

Amazon EC2

の永続的な Amazon EC2 インスタンスで分析を実行します AWS アカウント。このオプションは、ローカルマシンからコンピューティングをオフロードし、より大きな分析をサポートし、定期的なスケジュール分析を可能にします。インスタンスは送信間で実行され続けます。

エージェントスキルは、以下を含む AWS CloudFormation スタックをプロビジョニングします。

- Docker と継続的なモダナイゼーションコンテナを備えた Amazon EC2 インスタンス (Amazon Linux 2023)
- AWS Transform、Amazon S3、KMS、Secrets Manager、およびセキュリティエージェントのアクセス許可を持つ IAM ロール AWS
- AmazonSSMManagedInstanceCore SSM 経由のシェルアクセス用 (SSH キーペアまたはインバウンドポートは不要)
- インバウンドルールのないセキュリティグループ

IAM ユーザーまたはロールには、Amazon EC2 ライフサイクル管理、AWS CloudFormation スタックオペレーション、IAM ロール作成、Amazon S3 バケットオペレーション、Secrets Manager シークレット管理、SSM コマンドのアクセス許可が必要です。

EC2 実行を設定するには、Kiro Power またはエージェントプラグインをインストールし (「」を参照 [デベロッパーツール](#))、[「継続的なモダナイゼーション分析のために EC2 インスタンスを設定する」](#)とエージェントに依頼します。エージェントはインフラストラクチャをプロビジョニングし、コンテナが正常であることを確認し、SSM を介して分析を送信します。SSH は必要ありません。

AWS バッチ (Fargate)

サーバーレスコンピューティングのために、Fargate を使用して AWS Batch で分離ジョブとして分析を実行します。各分析は、永続的なインフラストラクチャを管理することなく、独自のコンテナで実行されます。複数のソースまたは分析タイプの並列分析に適しています。

このオプションは、AWS 変換 CDK インフラストラクチャスタックを再利用します。エージェントスキルは、以下を含む必要なインフラストラクチャをデプロイします。

- AWS バッチジョブキューとコンピューティング環境
- 継続的モダナイゼーションコンテナイメージを使用したジョブ定義
- バッチジョブ実行の IAM ロール

• ジョブ送信用の Lambda 関数

バッチ実行を設定するには、Kiro Power またはエージェントプラグインをインストールし (「」を参照 [デベロッパーツール](#))、エージェントに「Fargate で分析を実行する」または「継続的なモダナイゼーションのためにバッチ実行を設定する」と尋ねます。エージェントは CDK スタックをデプロイし、ソース認証情報を Secrets Manager に保存して、分析ジョブを AWS Batch に送信します。

セキュリティエージェントの設定

security 分析タイプは、脆弱性と CVE 検出に AWS Security Agent サービスを使用します。他の分析タイプとは異なり、では 1 回限りのインフラストラクチャ設定が必要です AWS アカウント。

setup コマンドは、以下を含む AWS CloudFormation スタックをプロビジョニングします。

- ソースコードのアップロード用の Amazon S3 バケット
- が引き受ける IAM ロール securityagent.amazonaws.com
- セキュリティエージェントロールの マネージドポリシー

セットアップを実行するには、IAM ユーザーまたはロールに次の追加のアクセス許可が必要です。

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

次のコマンドを使用して、セキュリティエージェントのインフラストラクチャを管理します。

```
# 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
```

セットアップが完了したら、他の分析タイプと同じ方法でセキュリティ分析を実行できます。

開始方法

前提条件

継続的なモダナイゼーションを効果的に使用するには、Git バージョン管理の基本的な理解、ソースコードプラットフォーム (GitHub、GitLab、または Bitbucket) AWS の知識、IAM アクセス許可の実務知識、コマンドラインツールの経験が必要です。

継続的なモダナイゼーションを使用する前に、以下があることを確認してください。

- AWS インストールされた変換 CLI:

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

- Node.js 22 以降
- 有効な AWS 認証情報 (AWS_PROFILE または環境変数)
- IAM ユーザーまたはロールにAWSTransformCustomFullAccessアタッチされた AWS マネージドポリシー

クイックスタート

1. 継続的なモダナイゼーションサーバーを起動します。

```
atx ct server &
```

2. ソースを追加する:

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

name はソースのわかりやすい識別子、*provider* はプラットフォーム (github、gitlab、bitbucket、または local)、*org-or-group* は組織名またはグループ名、#### は認証トークンです。

3. リポジトリの検出:

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

4. 分析を実行します。

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

5. 検出結果を調べる:

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

6. 検出結果の修正:

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

継続的なモダナイゼーションの使用

ソース管理

atx ct source コマンドを使用してリポジトリを接続します。サポートされているプロバイダー: GitHub、GitLab、Bitbucket、local。

GitHub 組織

トークン: repoスコープを持つ個人用アクセストークン (クラシック)。分析には読み取り専用、修復には完全なリポジトリ。

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

GitLab グループとユーザー

トークン: apiスコープを持つ個人用アクセストークン。

```
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
```

Bitbucket ワークスペースとプロジェクト

Bitbucket Cloud — スコープ:

read:repository:bitbucket、write:repository:bitbucket、read:pullrequest:bitbucket
また、--emailと も必要です--username。

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

Bitbucket データセンター:

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

ローカルリポジトリ

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

Important

--path は、単一のリポジトリではなく、git リポジトリをサブディレクトリとして含む親ディレクトリを指す必要があります。

ソースの管理

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

リポジトリの検出と管理

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

検出後:

```
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"
```

分析の実行

--type フラグは、実行する分析の種類を指定します。

- `rapid-techdebt-analysis` – 古い依存関係と簡単な成功。
- `tech-debt-comprehensive` – 依存関係、セキュリティ、パターン、パフォーマンス、保守性、アーキテクチャ、コード品質、インフラストラクチャの検出結果をカバーする、より詳細な AI を活用した分析。
- `security` – セキュリティの脆弱性と露出。
- `agentic-readiness` – AI エージェント用のリポジトリの準備状況 (フレームワーク、APIs、ドキュメント)。
- `modernization-readiness` – インフラストラクチャ、アプリケーション、データ、セキュリティ、運用の各側面にわたるモダナイゼーションの機会。

```
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]
```

カスタム分析

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

-g フラグ付き設定: key-value、JSON、またはファイルパス。

TDs一覧表示する: `atx custom def list`

検出結果の管理

```
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
```

検出結果のステータス

- open — アクティブ
- dismissed — 手動で却下 (理由が必要)
- obsolete — 再分析で結果が生成されなくなった場合のシステムセット

```
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
```

廃止の検出

再分析では、解決された検出結果は古いものとしてマークされます。再度開くことはできません。監査のために保持されます。

修復の作成

3 つのモード: 検出結果ベース、TD オーバーライド、直接 TD。

```
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
```

プロバイダー別の出力: GitHub PR、GitLab MR、Bitbucket PR、Local ブランチ。

Note

トークンには、PR/MR 作成用の書き込みアクセス権が必要です。

--local フラグ付きのローカル実行。

```
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 ウェブアプリケーションの変換 (オプション)

AWS Transform ウェブアプリケーションは、コードソースのポートフォリオ全体で継続的なモダナイゼーション分析、検出結果、修復をモニタリングするためのオプションのインターフェイスです。

AWS 変換ウェブアプリケーションを使用する前に、組織内の AWS 変換にアクセスするためのユーザー ID を有効にする必要があります。AWS 変換を有効にする方法については、[AWS「変換の設定」](#)を参照してください。

AWS Transform ウェブアプリケーションを使用するには:

1. AWS IAM Identity Center 認証情報を使用して にアクセスしてサインイン<https://aws.amazon.com/transform/>します。
2. サインイン後に継続的なモダナイゼーションが表示されない場合は、代わりに IAM 認証情報を使用してサインインします。IAM 認証情報のサインインを有効にするには:
 - a. AWS マネジメントコンソールで、AWS 変換を開き、設定を選択します。
 - b. 「IAM 認証情報を使用したアクセス AWS 変換」セクションで、IAM 認証情報アクセスを有効にします。
 - c. 同じ設定ページで、ウェブアプリケーション URL (IAM を使用) の下に表示されるサインインリンクをコピーします。
 - d. サインイン URL を、 マネジメントコンソールが開いているのと同じブラウザウィンドウに貼り付けます AWS 。これにより、ウェブアプリケーションはそのアカウントの AWS 認証情報を使用します。
3. 左側のナビゲーションメニューを開き、継続的なモダナイゼーションを選択します。ダッシュボードには、ソース、リポジトリ、重要度別の検出結果の合計、分析タイプなどのサマリー統計が表示されます。
4. 詳細な結果を表示するには、分析、結果、修復タブを使用します。
5. とチャットする ウェブアプリケーションから直接 AWS 変換して、分析、検出結果、または修復について質問します。

ウェブアプリケーションは、複数のコードベースにわたる継続的なモダナイゼーション分析と修復を一元的に可視化する必要があるエンタープライズ規模のオペレーション向けに設計されています。

デベロッパーツール

AWS Transform Power とエージェントプラグインを使用して、エージェント IDEs から継続的なモダナイゼーションにアクセスできます。エージェントは、会話による対話を通じて、継続的なモダナイゼーションのセットアップ、オンボーディング、継続的な使用を調整します。

- Kiro IDE — [AWS Transform Kiro Power](#) をインストールします。Kiro Chat を開き、継続的なモダナイゼーションワークフローを支援するようエージェントに依頼します。
- Claude Code、Codex、Cursor — [aws-labs/agent-plugins](#) (plugins/aws-transform ディレクトリ) をインストールします。エージェントプラグインは Kiro Power と同じ機能を提供します。

エージェントスキル

Kiro Power and Agent プラグインには、会話によるやり取りを通じて継続的なモダナイゼーションワークフローを調整するスキルが含まれています。

スキル	説明
ガイド	ワークフロー全体をステップバイステップで説明するインタラクティブなオンボーディング
ソース	ソース接続の追加、一覧表示、削除
発見	ソースをスキャンしてリポジトリを検出する
分析	すべてのタイプで分析を実行および管理します。
検出結果	分析結果のクエリ、フィルタリング、管理
修正	検出結果を修正するための修復を作成および管理します。
EC2 の実行	リモート分析用の Amazon EC2 インスタンスのプロビジョニングと管理
バッチ実行	Fargate を使用して分析ジョブを AWS バッチに送信する

スキル	説明
スケジュール	Amazon EventBridge スケジューラを使用して cron スケジュールで定期的な分析を設定する
報告	検出結果、重大度分布、修復の進行状況を示すグラフを含む自己完結型の HTML レポートを生成する

CLI リファレンス

次の表は、すべての `atx ct` サブコマンドをまとめたものです。

サブコマンド	説明
<code>atx ct server</code>	継続的なモダナイゼーションサーバーを起動する
<code>atx ct status</code>	ソース、リポジトリ、分析、検出結果、修復などのシステムステータスを表示する
<code>atx ct source add</code>	ソースを追加する (GitHub、GitLab、Bit bucket、またはローカル)
<code>atx ct source list</code>	設定済みソースを一覧表示する
<code>atx ct source remove</code>	ソースを削除する
<code>atx ct discovery scan</code>	ソースからリポジトリを検出する
<code>atx ct discovery status</code>	検出スキャンのステータスを確認する
<code>atx ct repository list</code>	オプションのフィルターを使用してリポジトリを一覧表示する
<code>atx ct repository get</code>	1 つのリポジトリを取得する
<code>atx ct repository update</code>	リポジトリラベルを更新する

サブコマンド	説明
<code>atx ct repository delete</code>	リポジトリを削除する
<code>atx ct analysis run</code>	リポジトリで分析を実行する
<code>atx ct analysis get</code>	分析の詳細を取得する
<code>atx ct analysis list</code>	オプションのステータスフィルターとタイプフィルターを使用して分析を一覧表示する
<code>atx ct analysis cancel</code>	実行中の分析をキャンセルする
<code>atx ct analysis delete</code>	分析を削除する
<code>atx ct findings list</code>	リポジトリ、ソース、重要度、タイプ、ステータス、分析のフィルターを使用して結果を一覧表示する
<code>atx ct findings get</code>	1 つの検出結果を取得する
<code>atx ct findings update</code>	検出結果のステータスを更新する (オープンまたは却下)
<code>atx ct findings batch-update</code>	複数の検出結果をバッチ更新する
<code>atx ct findings delete</code>	却下された検出結果または廃止された検出結果を削除する
<code>atx ct remediation create</code>	検出結果または変換定義から修復を作成する
<code>atx ct remediation list</code>	すべての修復を一覧表示する
<code>atx ct remediation status</code>	修復ステータスを確認し、PR/MR リンクを表示する
<code>atx ct remediation retry</code>	失敗した修復を再試行する
<code>atx ct remediation delete</code>	修復を削除する

サブコマンド	説明
<code>atx ct setup security-agent</code>	セキュリティエージェントのインフラストラクチャをプロビジョニングまたは管理する
<code>atx ct schema</code>	すべてのコマンド、オプション、引数など、完全な <code>atx ct</code> コマンドサーフェスの機械読み取り可能な JSON マニフェストを印刷して、エージェントとオートメーションがヘルプテキストを解析せずに CLI を検出できるようにします。

トラブルシューティング

接続が拒否されたか、サーバーが実行されていません

継続的なモダナイゼーション CLI には、実行中のサーバーが必要です。で開始 `atx ct server` &し、 で検証します `atx ct status --health`。

検出スキャンがリポジトリをゼロに返す

ローカルソースの場合、 がリポジトリを直接 (など `/home/user/repos`) ではなく、サブディレクトリとしてリポジトリを含む親ディレクトリ (など) を `--path` 指していることを確認します `/home/user/repos/my-app`。スキャナーは、 `.git` フォルダを含む子ディレクトリを検索します。

SETUP_REQUIRED エラー

ソースはアカウントに存在しますが、認証情報はこのマシンで設定されていません。を実行して、認証情報 `atx ct source add --name name` をローカルで設定します。

AUTH_REQUIRED エラー

ソースに有効なトークンが見つかりませんでした。トークンが正しく、必要なスコープがあることを確認します。GitHub の場合、トークンに `repo` スコープがあることを確認します。GitLab の場合は、 `api` スコープを確認します。

INVALID_INPUT エラー

正しい分析タイプ名を使用していることを確認します。有効な値は、rapid-techdebt-analysis、tech-debt-comprehensive、security、agentic-readiness、modernization-readiness、custom です。

修復中のアクセス許可エラー

修復により、ブランチとプル/マージリクエストが作成され、リポジトリへの書き込みアクセスが必要になります。トークンを書き込みアクセス許可で更新します。GitHub の場合は、フルrepoスコープを確認してください。GitLab の場合は、apiスコープを確認します。Bitbucket の場合は、write:repository:bitbucketと write:pullrequest:bitbucketスコープを確認します。

セキュリティエージェントの設定が失敗する

IAM ユーザーまたはロールに AWS CloudFormation、 のセキュリティエージェントセットアップセクションに記載されている必須の IAM および Amazon S3 アクセス許可があることを確認します [AWS Transform の継続的なモダナイゼーションの仕組み](#)。でステータスを確認します `atx ct setup security-agent --status`。

継続的なモダナイゼーションがウェブアプリケーションに表示されない

Transform ウェブアプリケーションにサインインした後に継続的なモダ AWS ナイゼーションが表示されない場合は、IAM Identity Center ではなく、AWS Transform が有効になってい AWS アカウント の IAM AWS 認証情報を使用してサインインします。手順については、「[AWS ウェブアプリケーションの変換 \(オプション\)](#)」を参照してください。

検出ツール

AWS Transform 検出ツールは、組織内のサーバーインベントリを自動的に検出して移行に備えるのに役立ちます。このツールは、VMware vCenter、Microsoft Hyper-V、およびインポートされたサーバーをサポートしています。検出ツールを使用するには、1 つ以上の検出ソースを設定し、ツールを実行させ、検出済みインベントリペインで結果を確認します。

検出ソースを設定すると、検出ツールは情報の収集を開始します。検出ツールがネットワークを完全に分析するために必要な時間は、環境のサイズによって異なります。指向性移行のビジネスケースでは、サーバー収集の完了後にツールが生成する移行ポートフォリオ評価 (MPA) ファイルを使用できます。

検出ツールのワークフロー

検出ツールのワークフローは、次の 2 種類のアクティビティで構成されます。

- 設定アクティビティ
- データのレビューと使用

次の手順では、検出ツールをインストールして使用し、収集されたデータを操作する方法について説明します。

1. 検出ツールをインストールします。
2. 検出ソースを設定します。CSV ファイルインポートを使用して、VMware vCenter、Microsoft Hyper-V ホスト、またはサーバーを 1 つ以上設定できます。データ検出は、ソースを設定した後に開始されます。
3. OS アクセスを設定し、サーバー、データベース、ネットワーク接続の収集ステータスを確認します。
 - 必要に応じて OS 認証情報を調整します。
4. (オプション) Oracle 認証情報を設定して直接 SQL 収集を有効にするか、SSH または WinRM による OS レベルのフォールバック検出を使用します。
5. 移行ビジネスケースを生成するには、.zip ファイルを[移行評価](#)にアップロードするか、解凍して mpa_exports ディレクトリから MPA ファイルをアップロードします。エクスポートには、設定されたすべてのソースからのデータが含まれ、VMware、Hyper-V、およびインポートされたサーバーの MPA ファイルが含まれます。

検出ツールは、次の検出パスをサポートしています。

- VMware vCenter 自動検出 – VMware vCenter によって管理されるサーバーを自動的に検出します。
- Hyper-V 自動検出 – Microsoft Hyper-V ホストによって管理されるサーバーを自動的に検出します。
- サーバーのインポート – CSV ファイルを使用してサーバーインベントリを手動でインポートします。

検出ツールのセットアップ

前提条件

AWS Transform 検出ツールを使用するための前提条件は次のとおりです。

一般的な前提条件

- このツールには、4 vCPU、16 GB の RAM、35 GB のハードディスクが必要です。
- DHCP は、検出ツール VM のネットワークで利用できる必要があります。
- このツールは、一元化されたアプローチを使用してデータを収集します。スコープ内のサーバーは、検出ツール VM からのインバウンド接続を許可する必要があります (デフォルトポート、カスタムポート設定がサポートされています)。
 - Linux – SSH TCP/22
 - Windows – HTTP の場合は TCP/5985、HTTPS の場合は TCP/5986
 - SNMP – UDP/161 (OS メトリクスではなくネットワーク収集にのみ使用)
- Linux の場合、SSH を使用してサーバーに接続できるユーザーアカウント。検出ツールは、ネットワークコレクションと OS メトリクスのコマンドを SSH 経由で実行します。ほとんどのコマンドは通常のユーザーとして実行されます。(サーバー UUID と製造元)、dmidecode (LVM 検出)、ss または lvs (プロセス情報を含む netstat ネットワーク接続) の少数のコマンドが sudo を試行し、sudo が使用できない場合は自動的にフォールバックします。sudo を使用しない場合、検出ツールは依然としてほとんどのデータを収集しますが、UUID、LVM、およびプロセスレベルのネットワークの詳細は失われます。完全なデータ収集を確保するために、SSH ユーザーにパスワードレス sudo を設定することをお勧めします。詳細な内訳については、「」を参照してください [the section called “必要なアクセス許可”](#)。

VMware の前提条件

- VMware vCenter Server バージョン 6.5、6.7、7.0、または 8.0。
- VMware vCenter に OVA をデプロイするアクセス許可。
- VMware vCenter Server のセットアップの場合、ルートデータセンターレベルで読み取り専用ロールが割り当てられた vCenter ユーザー。詳細については、「[the section called “必要なアクセス許可”](#)」を参照してください。

Hyper-V の前提条件

- Hyper-V ロールが有効になっている Windows Server。
- Hyper-V ホストで有効になっている WinRM。
- 各 Hyper-V ホストのリモート管理ユーザー、Hyper-V 管理者、および Performance Monitor ユーザーグループのメンバーであり、root\cimv2 名前空間への WMI 読み取りアクセス権を持つユーザーアカウント。詳細については、「[the section called “必要なアクセス許可”](#)」を参照してください。
- サポートされている認証: NTLM (HTTPS のみ) および Kerberos (HTTP または HTTPS)。

サーバーインポートの前提条件

- サーバーホスト名または IP アドレスと、検出ツールで設定または設定される OS 認証情報のフレンドリ名にマッピングされる認証情報名 (オプション) を含む CSV ファイル。CSV では、次のヘッダーを使用する必要があります。

```
hostname_or_ip,os_credential_name,oracle_credential_name
```

- サーバーは、検出ツール VM から到達可能である必要があります。デフォルトポート: Linux の場合は SSH ポート 22、Windows の場合は WinRM ポート 5985/5986。カスタムポートがサポートされています。

Linux インストーラの前提条件

- サポートされている Linux ディストリビューション: Amazon Linux 2、Amazon Linux 2023、RHEL 8~9、Rocky Linux 8~9、AlmaLinux 9、Ubuntu 20.04~24.04、Debian 10~12、または SLES 15 SP5。
- 最小 4 vCPU、16 GB RAM、35 GB の使用可能なディスク容量。

- ポート 5000 が使用可能である必要があります (別のサービスでは使用できません)。
- systemd はサービス管理に必要です。
- 検出ツール VM には、ターゲットインフラストラクチャへのネットワークアクセスが必要です。
デフォルトポート: ポート 443 の vCenter、ポート 5985/5986 の Hyper-V ホスト、ポート 22 のサーバー。カスタムポートは、SSH、WinRM、SNMP でサポートされています。

Oracle Database の前提条件

- 検出ツールからポート 1521 (またはカスタムポート) 上の Oracle ホストへのネットワークアクセス。
- サポートされている Oracle バージョン: Oracle Database 12c Release 1 (12.1) 以降は、直接 SQL 接続を使用して収集できます。OS レベルのフォールバック検出は、すべての Oracle バージョンで機能します。
- SELECT_CATALOG_ROLE 許可を持つ読み取り専用の Oracle サービスアカウント。詳細については、「[the section called “Oracle データベース \(SQL\)”](#)」を参照してください。
- OS レベルのフォールバック検出の場合: Oracle ホストへの SSH または WinRM アクセス (既存の OS 認証情報を使用)。

VMware にデプロイする

VMware 仮想マシンの仕様

- オペレーティングシステム – Amazon Linux 2023
- RAM – 16 GB
- CPU – 4 コア
- ディスク – 35 GB
- VMware の要件 – [VMware で AL2023 を実行するための VMware ホスト要件](#)」を参照してください。

VMware OVA をデプロイする

1. 次の URL から OVA ファイルをダウンロードします: <https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.ova>
2. VMware 管理者として vCenter にサインインします。

3. OVA ファイルをインストールするには、次のいずれかの方法を使用します。

- a. UI を使用する: ファイルを選択し、OVF テンプレートのデプロイを選択し、ステップ 1 でダウンロードした検出ツールの OVA ファイルを選択して、ウィザードを完了します。サーバー管理ダッシュボードのプロキシ設定が正しく設定されていることを確認します。
- b. コマンドラインを使用する: コマンドラインから検出ツール OVA ファイルをインストールするには、VMware Open Virtualization Format Tool (ovftool) をダウンロードして使用します。ovftool をダウンロードするには、[OVF ツールドキュメント](#) ページからリリースを選択します。これは、ovftool コマンドラインツールを使用して検出ツール OVA ファイルをインストールする例です。

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

この例の置き換え可能な値の説明:

- 名前は、検出ツール VM に使用する名前です。
 - データストアは、vCenter 内のデータストアの名前です。
 - OVA ファイル名は、ダウンロードした検出ツール OVA ファイルの名前です。
 - ユーザー名/パスワードは vCenter 認証情報です。
 - vcenterurl は vCenter の URL です。
 - vi パスは、VMware ESXi ホストへのパスです。
4. vCenter でデプロイされた検出ツールを見つけます。VM を右クリックし、Power、Power On を選択します。
 5. 数分後、検出ツールの IP アドレスが vCenter に表示されます。この IP アドレスを使用して検出ツールに接続します。

Hyper-V にデプロイする

Hyper-V 仮想マシンの仕様

- オペレーティングシステム – Amazon Linux 2023
- RAM – 少なくとも 16 GB を割り当てることをお勧めします
- CPU – 少なくとも 4 コアを割り当てることをお勧めします

- ディスク – 35 GB (VHD に含まれる)
- Hyper-V 要件 – [Hyper-V で AL2023 を実行するための Hyper-V ホスト要件](#)を参照してください。

Hyper-V VHD をデプロイする

1. この URL から VHD ファイルをダウンロードします: <https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.vhd>
2. Hyper-V ロールが有効になっている Windows Server マシンに VHD ファイルをコピーします。
3. Hyper-V Manager を開きます。
4. 「新規」を選択し、「仮想マシン」を選択します。
5. セットアップウィザードを完了します。生成を指定ページで、生成 1 を選択します。第 2 世代仮想マシンは VHD 形式をサポートしていません。メモリの割り当てページで、少なくとも 16384 MB を割り当てます。仮想ハードディスクの接続ページで、既存の仮想ハードディスクの使用を選択し、コピーした VHD ファイルを選択します。
6. VM を起動します。数分後、Hyper-V Manager の VM のネットワークタブで IP アドレスを確認するか、VM コンソールに接続して `ip addr` を実行します。この IP アドレスを使用して検出ツールに接続します。

Linux にデプロイする

検出ツールは、サポートされている Linux ディストリビューションにデプロイするためのスタンドアロン Linux インストーラとして使用できます。このオプションは、仮想マシンをデプロイするのではなく、検出ツールを既存の Linux サーバーに直接インストールする場合に使用します。

Linux サーバーの仕様

- RAM – 少なくとも 16 GB を割り当ててをお勧めします
- CPU – 少なくとも 4 コアを割り当ててをお勧めします
- ディスク – 35 GB

サポートされているディストリビューション

- Amazon Linux – Amazon Linux 2、Amazon Linux 2023
- Red Hat Enterprise Linux および派生 – RHEL 8~9、Rocky Linux 8~9、AlmaLinux 9

- Ubuntu – 20.04、22.04、24.04
- Debian – 10、11、12
- SUSE – SLES 15 SP5

検出ツールをインストールする

次の URL からインストーラスクリプトをダウンロードします: <https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.sh>

Linux に検出ツールをインストールするには

1. インストーラスクリプトを Linux サーバーにダウンロードします。

```
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. (オプション) 互換性チェックを実行して、システムが要件を満たしていることを確認します。

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

3. 検出ツールをインストールします。以下のオプションのいずれかを選択してください。

オプション 1: 自動依存関係インストールで をインストールする (推奨)

このオプションは、サービスをインストールする前に、欠落しているシステム依存関係を自動的にインストールします。

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

オプション 2: 自動依存関係のインストールなしで をインストールする

このオプションは、依存関係がないかどうかをチェックし、重要な依存関係がない場合に手動インストール手順でエラーを表示します。組織が自動パッケージインストールを制限し、システムパッケージを手動で管理する場合は、このオプションを使用します。

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

4. 検出ツールサービスを起動します。

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

5. で検出ツールにアクセスしますhttps://ip_address:5000。

検出ツールをアンインストールするには、を実行します `sudo ./AWS-Transform-discovery-tool.sh uninstall`。

次のコマンドを使用してサービスを管理することもできます。

- `sudo ./AWS-Transform-discovery-tool.sh status` – サービスが実行されているかどうかを確認します。
- `sudo ./AWS-Transform-discovery-tool.sh stop` – サービスを停止します。
- `sudo ./AWS-Transform-discovery-tool.sh logs` – サービスログを表示します。

システムの依存関係

インストーラは、次のシステム依存関係をチェックします。

依存関係	緊急度	目的
OpenSSL	重大	データベース暗号化キーを生成します
libcrypt	重大	Python ランタイムで必要
Kerberos ライブラリ	警告	Windows サーバーへの Kerberos 認証に必要です。これがないと、NTLM と基本認証は引き続き機能します。

検出ツールを開始するには、重要な依存関係が必要です。それらが欠落していて、オプション 2 (なし `--install-deps`) を選択した場合、インストーラはエラーを表示し、手動でインストールする手順を示します。警告レベルの依存関係はオプションです。インストーラは警告しますが、インストールを続行します。

Note

インストーラは、glibc バージョンや systemd の可用性など、システムの互換性をチェックします。systemd 250 以降 (Amazon Linux 2023、RHEL 9、Rocky 9、AlmaLinux 9、Ubuntu

24.04+、Debian 12+) を使用するシステムでは、データベース暗号化キーは systemd-creds を使用して保管時に暗号化されます。古いシステムでは、暗号化キーはアクセス許可で保護されたファイルとして保存されます。

検出ツールを設定する

検出ツールコンソールにアクセスする

1. ウェブブラウザアクセス `https://ip_address:5000` の場合: *#ip_address* は Deploy Discovery Tool の検出ツールの IP アドレスです。検出ツールは、HTTPS 接続に自己署名証明書を使用するため、セキュリティ警告が発生します。リスクを受け入れるを選択し、引き続き検出ツールコンソールに進みます。
2. 検出ツールコンソールに初めてアクセスする場合は、検出ツールのログインパスワードを作成します。今後のログインに使用するパスワードを作成します。

Important

このパスワードを覚えておいてください。パスワード復旧メカニズムはありません。

検出ツール VM へのアクセス

- 検出ツール VM には、デフォルトでユーザー名とパスワード (「検出」、「パスワード」) が付属しています。セキュリティを強化するために、ハイパーバイザーのコンソール (VMware 用 vSphere Client や Hyper-V 用 Hyper-V Manager など) から VM にログイン `sudo passwd discovery` した後、 を使用してパスワードを更新することをお勧めします。
- SSH アクセスはデフォルトで無効になっています。ユーザーは、事前設定済みの `disablesssh` エイリアス `enablesssh` とエイリアスを使用して、検出ツール VM への SSH アクセスを有効または無効にできます。ユーザーは、SSH アクセスを有効に `ssh discovery@<VM-IP>` した後、 を介して VM に SSH 接続できます。ユーザーは、SSH アクセスをほとんどの場合無効にし、アクティブに必要な場合にのみ有効にすることをお勧めします。パスワードの変更は、 の実行時に適用されます `enablesssh`。
- で検出ツールのデータディレクトリにアクセスするには `/home/ec2-user/.local/share/DiscoveryTool`、 `ec2-user` を実行して に切り替えることをお勧めします `sudo su ec2-user`。

Kerberos 認証の設定

Kerberos 認証は、検出ツールから Windows サーバーに接続するために推奨される方法です。検出ツール VM は、ネイティブ Amazon Linux 2023 Kerberos ライブラリを使用して Active Directory ドメインに対して認証します。

以下は、検出ツール VM での Kerberos 認証に関する重要なポイントです。

- kinit コマンドを使用して Kerberos チケットを取得し klist、チケットを検証します。
- Kerberos 設定ファイルは にあります/etc/krb5.conf。
- 検出ツールを設定する前に、 が検出ツール VM の CLI からkinit成功することを確認します。

Kerberos の前提条件

Kerberos 認証を設定する前に、次の情報とネットワーク接続があることを確認してください。

1. Active Directory 管理者から次の情報を取得します。
 - Kerberos 領域名 (通常は など、大文字のドメイン名EXAMPLE.COM)。
 - Key Distribution Center (KDC) のホスト名または IP アドレス。通常はドメインコントローラーです (例: dc01.example.com)。
 - ターゲット Windows サーバーに対して認証するアクセス許可を持つサービスアカウント。
2. 検出ツール VM が以下とネットワーク接続していることを確認します。
 - Kerberos 認証用のポート 88 (TCP および UDP) の KDC。
 - WinRM ポートのターゲット Windows サーバー (HTTP の場合は 5985、HTTPS の場合は 5986)。

Kerberos の設定

検出ツール VM で Kerberos 認証を設定するには、次の手順を実行します。

1. 検出ツール VM への SSH。

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

2. で Kerberos 設定ファイルを編集します/etc/krb5.conf。

```
sudo nano /etc/krb5.conf
```


次の設定を追加し、ブレースホルダー値を環境の詳細に置き換えます。

```
[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 では大文字と小文字が区別されます。領域名は大文字にする必要があります (例: ではなく EXAMPLE.COMexample.com)。[domain_realm] セクションのドメイン名は小文字にする必要があります。

複数の Active Directory ドメイン

検出ツールは、さまざまな Active Directory ドメインに対して複数の Kerberos 認証情報をサポートします。各認証情報は個別に認証されるため、複数の認証情報を正常に設定でき、分離は自動的に行われます。

複数のドメインにサーバーがある場合は、/etc/krb5.confファイル内の各領域にエントリを追加します。

```
[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. kinit コマンドを実行して、Kerberos チケットを取得できることを確認します。

```
kinit username@REALM.COM
```

プロンプトが表示されたら、パスワードを入力します。コマンドがエラーなしで完了した場合、認証は成功しました。

4. klist コマンドを実行してチケットを検証します。

```
klist
```

予想される出力は次のようになります。

```
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. で使用したのと同じ大文字と小文字を区別するプリンシパルを使用して検出ツールを設定します
kinit (例: username@REALM.COM)。

環境の DNS SRV レコードが Kerberos サービス検出用に設定されている場合、明示的な krb5.conf 設定は必要ありません。Kerberos 設定オプションの詳細については、[MIT Kerberos krb5.conf ドキュメント](#)と[サンプル krb5.conf ファイル](#)を参照してください。

ドメイン結合マシンから Kerberos 設定を検索する

Kerberos 設定の詳細がない場合は、ドメインに参加している Windows マシンから取得できます。ドメインに参加しているマシンのコマンドプロンプトから次のコマンドを実行します。

ドメイン名を検索するには、次のコマンドを実行します。

```
echo %USERDNSDOMAIN%
```

出力の例:

```
EXAMPLE.COM
```

ドメインコントローラーのホスト名を検索するには、次のコマンドを実行します。

```
nltest /dsgetdc:EXAMPLE.COM
```

出力の例:

```
DC: \\dc01.example.com
Address: \\10.0.1.100
Dom Guid: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
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
```

krb5.conf 次のように出力を設定にマッピングします。

- 領域 – 大文字%USERDNSDOMAIN%の の値を使用します (例: EXAMPLE.COM)。
- KDC – nltest出力から DC ホスト名を使用します (例: dc01.example.com)。

vCenter アクセスを設定する

1. 検出ツールページのステップ 1。検出ソースを設定し、ソースの設定を選択します。
2. 検出ソースの設定ページで、フレンドリ名、vCenter FQDN/IP、ユーザー名、パスワードを指定します。
3. [設定の保存] を選択します。

検出ツールは、[「検出済みインベントリ」](#)で説明されているように、vCenter 情報の収集を開始します。

初期設定後、検出ツールのステータスフレームで vCenter アクセスの編集 を選択して、vCenter アクセス設定を変更します。

検出ツールは、設定されたすべての vCenter サーバーから並行して収集します。収集中に 1 つの vCenter サーバーにアクセスできない場合、ツールは部分的な成功を報告し、残りの vCenter サーバーからの収集を続行します。

VM が複数の vCenter サーバーに表示される場合 (共有 ESXi ホストやクロス vCenter vMotion など)、検出ツールは VMs。各一意の VM はインベントリに 1 回のみ表示されます。

Hyper-V アクセスを設定する

1. 検出ツールページのステップ 1。検出ソースを設定し、ソースの設定を選択します。
2. 検出ソースの設定ページで、わかりやすい名前、ホスト FQDN または IP アドレス、認証タイプ (NTLM または Kerberos)、WinRM ユーザー名、および WinRM パスワードを指定します。
3. [設定の保存] を選択します。

検出ツールは、[「検出済みインベントリ」](#)で説明されているように、Hyper-V 情報の収集を開始します。

認証情報を保存すると、収集が自動的に開始されます。

Hyper-V フェイルオーバークラスターの場合、同じクラスターに複数のホストを追加できます。このツールは、複数のホストに表示される VMs を自動的に重複排除します。

サーバーのインポート

1. 検出ツールのホームページからサーバーのインポートページに移動します。
2. (必須)、hostname_or_ip (os_credential_name オプション)、oracle_credential_name (オプション) の列を含む CSV ファイルを準備します。
 - hostname_or_ip 値は有効な IPv4 アドレスまたは完全修飾ドメイン名 (FQDN) である必要があります。
 - os_credential_name 値を指定する場合、設定済みの OS 認証情報 (SSH、WinRM、または SNMP) のわかりやすい名前と一致する必要があります。OS 認証情報をまだ設定していないサーバーの場合は空のままにします。
 - oracle_credential_name 値を指定する場合、設定済みの Oracle 認証情報のわかりやすい名前と一致する必要があります。

3. CSV ファイルをアップロードします。このツールはすべての行を検証し、いずれかの行が無効である場合はファイルを拒否します。

インポートが成功すると、OS 認証情報が設定されている場合、ツールはインポートされたサーバーのデータベース、ネットワーク、OS メトリクスの収集を自動的に開始します。別の CSV ファイルをアップロードすると、既存のレコードは重複することなく更新され、新しいレコードはインベントリにマージされます。

自己署名認証機関を検出ツールにインポートする (オプション)

これは、自己署名認証機関 (CA) によって署名された WinRM HTTPS 証明書を使用して HTTPS 経由で WinRM を使用し、検出ツールで「サーバー SSL 証明書の検証」を有効にする場合に必要です。

前提条件

1. ターゲットサーバーで WinRM HTTPS 証明書に署名するために使用された自己署名 CA 証明書
2. PEM 形式の証明書 (.pem または .crt 拡張機能)

検出ツール VM に自己署名認証機関をインポートするには:

1. 検出ツール VM への Ssh
2. ターゲットサーバーの WinRM 証明書に署名した CA 証明書 (複数可) を、検出ツール VM / `etc/pki/ca-trust/source/anchors/` の信頼ストアディレクトリに配置します。例: `sudo cp winrm-ca.pem /etc/pki/ca-trust/source/anchors/winrm-ca.pem`。注: ターゲットサーバーが異なる CAs によって署名された証明書を使用する場合は、関連するすべての CA 証明書をこのディレクトリにコピーします。
3. 証明書トラストストアを更新します。 `sudo update-ca-trust`
4. VM を再起動する
5. (オプション) 証明書が正常にインポートされたことを確認するには、次のコマンドを実行します。 `sudo trust list -filter=ca-anchors | grep -A 5 "<certificate_name>"`

[「Windows リモート管理のインストールと設定」](#) を参照してください。

OS アクセス用の検出ツールを設定する

検出ツールが以下を実行できるように OS アクセスを設定します。

- データベースを評価し、VM の移行を支援するデータベースを検出します。
- 各接続に関連付けられたプロセスなど、インベントリ内のサーバー間のネットワーク接続を追跡して、アプリケーションの依存関係マッピングとウェーブプランニングを支援します。両方のエンドポイントが検出ツールのインベントリにある接続のみが含まれます。

検出ツールの OS アクセスを有効にする

1. OS アクセスの設定ページに移動して、Windows および Linux の認証情報を指定します。
2. 認証情報を追加するプロトコルを選択します。
3. 選択したプロトコルに必要な認証情報を入力します。
4. 自動接続を選択すると、検出ツールは、各サーバーに一致する認証情報が見つかるまで、検出されたサーバーで提供されたすべての認証情報を試すことができます。

自動接続機能に関する重要なセキュリティ推奨事項 [the section called “注意して自動接続機能を使用する”](#) については、「」を参照してください。

5. セットアップと接続を選択します。

OS マッチングプロセスが完了すると、データ収集が進行中であることを示すメッセージと、認証情報が一致するサーバーに関するエラーが表示されます。

サポートされているプロトコルの設定

検出ツールがターゲットサーバーと通信するには、ターゲットサーバーで WinRM、SSH、SNMP プロトコルを設定する必要があります。

WinRM と WMI のセットアップ

WinRM は、現在サポートされているすべてのバージョンの Windows オペレーティングシステムに自動的にインストールされます。

WinRM 設定を確認または編集するには、`winrm` コマンドラインツールを使用します。

- インストールされている WinRM リスナーを確認します。 `winrm enumerate winrm/config/listener`
- WinRM 設定を確認します。 `winrm get winrm/config`
- WinRM をセットアップするコマンドの例: `winrm quickconfig -transport:https`

リスナーポート

デフォルトの HTTP ポートは 5985、HTTPS は 5986 です。必要に応じて他のポートを使用できます。ポートは、検出ツールとターゲットサーバーの間に開いている必要があります。

暗号化

検出ツールは、暗号化された WinRM 通信を使用します。ターゲットサーバーの WinRM リスナーも暗号化を使用することをお勧めします。winrm set winrm/config/service '@{AllowUnencrypted="false"}'

NTLM と Kerberos

WinRM 認証プロトコル Kerberos と NTLM は、検出ツールでサポートされています。NTLM は HTTPS でのみ使用でき、Kerberos は HTTP または HTTPS の両方で使用できます。

WMI の要件

検出ツールは、次の WMI 名前空間をクエリします。WinRM アカウントには、コレクションモジュールに関連する各名前空間への読み取りアクセスが必要です。

WMI 名前空間	用途:
root\cimv2	OS メトリクス、Hyper-V ホストメタデータ、SQL Server コレクション
root\virtualization\v2	Hyper-V VM インベントリ
root\StandardCIMV2	ネットワークコレクション
root\Microsoft\SqlServer\ComputerManagement*	SQL Server コレクション
root\Microsoft\SqlServer\ReportServer*	SQL Server コレクション (SSRS)

ネットワーク収集では、以下の条件が満たされていることを確認します。

- ICMP を介したネットワーク接続を許可する
- TCP ポート 135 + 一時 TCP ポート範囲 (49152 ~ 65535) を介したネットワーク接続を許可する

- UAC を無効にする
- リモート DCOM アクセス許可が設定されている
- 最小限のアクセス許可で専用サービスアカウントを作成する
- WMI 名前空間のアクセス許可は\\root\\standardcimv2、名前空間、MSFT_NetTCPConnection クラスを持つ Windows アカウントに設定されます。

SQL Server コレクションでは、複雑な WMI オブジェクトのアクセス許可要件のため、ローカル管理者グループに属する Windows アカウント (ローカルまたはドメイン) が必要です。

SSH をセットアップする

- デフォルトのポートは 22 です。カスタムポートがサポートされています。設定済みのポートは、検出ツールとターゲットサーバーの間で開いている必要があります。
- SSH ネットワークコレクションが正しく機能するには、パスワードレス sudo 用に設定されたユーザーを指定します。
- ターゲット Linux サーバー (ほとんどのディストリビューションにデフォルトでインストール) で次のコマンドが使用可能であることを確認します。ネットワークコレクションssnetstatの場合はlsblkkiostat、OS メトリクスコレクションdfの場合は、、dmidecode、smartctltop、psfree、ip、、および。

検出ツールは、SSH の 2 つの認証方法をサポートしています。

オプション 1: ユーザー名とパスワード

SSH ユーザー名とパスワードを入力します。これはデフォルトの認証方法です。

オプション 2: SSH プライベートキー

SSH ユーザー名とプライベートキーを PEM 形式で指定します。このオプションを使用するには、SSH 認証情報を設定するときに認証タイプのドロップダウンから SSH キーを選択します。プライベートキーがパスフレーズで暗号化されている場合は、オプションのキーパスフレーズフィールドにパスフレーズを入力します。

次のキー形式がサポートされています。

- RSA
- ECDSA
- Ed25519

- OpenSSH 形式
- PKCS#8 形式

どちらの認証方法も自動接続をサポートしています。認証情報は保管時に暗号化されて保存されます。

SNMP のセットアップ

- デフォルトのポートは 161/UDP です。カスタムポートがサポートされています。設定済みのポートは、検出ツールとターゲットサーバーの間で開いている必要があります。
- SNMP v2 の場合: TCP 接続 OIDs。
- SNMP v3 の場合: TCP 接続 OIDs にアクセスできる読み取り専用アクセス許可を持つユーザー名/パスワードと認証/プライバシーの詳細を指定します。

検出ツールでは、以下にアクセスする必要があります。

- "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)

Oracle データベースアクセスを設定する

SQL 接続を介して詳細な Oracle データベースメタデータを直接収集するように Oracle データベースアクセスを設定します。収集されるメタデータには、CDB および PDB トポロジ、機能の使用、インストールされたオプションが含まれます。このデータは、Oracle データベースの移行をより正確に計画するのに役立ちます。Oracle Database 12c Release 1 (12.1) 以降は、直接 SQL 接続を介して収集できます。OS レベルのフォールバック検出は、すべての Oracle バージョンで機能します。

検出ツールで Oracle 認証情報を設定する

1. 検出ツールページのサイドバーで、データベースアクセスを選択します。
2. Oracle 認証情報の追加を選択します。
3. 以下の情報を指定します。
 - わかりやすい名前 – この認証情報のわかりやすい名前 (例: Oracle Production)。
 - ポート – Oracle リスナーポート (デフォルトは 1521)。

- サービス名 – ターゲットデータベースの Oracle サービス名。
 - Username – Oracle サービスアカウントのユーザー名。
 - パスワード – Oracle サービスアカウントのパスワード。
 - 自動接続 — このオプションをオンにして、インベントリ内のすべてのサーバーに対して認証情報を試します。認証情報を特定のサーバーに手動で割り当てるには、このオプションをオフにします。
4. 認証情報をさらに追加するには (異なる Oracle 環境など)、Oracle 認証情報を再度追加を選択します。
 5. [保存] を選択します。

認証情報モード

Oracle 認証情報を設定するときは、次の 2 つのモードから選択できます。

- 手動 – 認証情報を特定のサーバーに固定します。検出ツールは、そのサーバーにのみその認証情報を使用します。接続が失敗した場合、フォールバックは発生しません。認証情報設定を修正して問題を解決します。
- 自動接続 – 検出ツールは、インベントリ内のすべてのサーバーに対して各自動接続認証情報を試行します。サーバーの認証情報が成功すると、検出ツールはそれ以降のすべてのコレクションラウンドでその認証情報を使用します。

検出フロー

Oracle 認証情報を設定すると、検出ツールはまず直接 SQL 接続を試行します。すべてのデータベース認証情報が失敗した場合、ツールは SSH または WinRM による OS レベルの検出にフォールバックするため、データベースにアクセスせずに Oracle のインストールを検出できます。

検出ツールの更新

検出ツールには自動更新機能はありませんが、更新するインストールから 30 日後にリマインダー通知が送信されます。最新の機能とセキュリティパッチを受け取るには、アプリケーションを up-to-date 状態に保つことをお勧めします。

ツールを手動で更新するには

1. 提供されたリンクから最新の検出ツールイメージファイル (OVA for VMware または VHD for Hyper-V) をダウンロードします。

2. (オプション) 最新の検出ツールイメージファイルをデプロイする前に、前の検出ツールイメージファイルを削除することをお勧めします。
3. 「検出ツールのデプロイ」セクションの手順に従って、更新されたバージョンをデプロイします。

アクセスの取り消し

検出ソースごとに個別にアクセスを取り消すことができます。1 つのソースのアクセスを取り消すと、他のソースからのデータは影響を受けません。

- vCenter アクセスの取り消し – vCenter 認証情報と VMware が収集したデータを削除します。Hyper-V データ、インポートされたサーバーデータ、または OS 認証情報は削除されません。
- Hyper-V アクセスの取り消し – Hyper-V 認証情報と Hyper-V-collected データのみを削除します。
- インポートされたサーバーの削除 – インベントリからインポートされたサーバーを削除します。これらのサーバーから収集されたダウンストリーム収集データ (ネットワーク、データベース) は保持されます。

検出ツールに必要なアクセス許可

検出ツールは、複数のプロトコルを使用してインフラストラクチャに接続します。各プロトコルとコレクションモジュールには、特定のアカウントのアクセス許可が必要です。このセクションでは、完全なデータ収集に必要な最小限のアクセス許可と、特定のアクセス許可が利用できない場合に失うデータについて説明します。

VMware vCenter

検出ツールは、ポート 443 (HTTPS) で VMware vCenter Server に接続し、読み取り専用オペレーションを実行します。vCenter 環境は変更されません。

最小必須ロール: vCenter ルートレベルで割り当てられた読み取り専用。

読み取り専用ロールは、検出ツールが VM インベントリ (名前、UUID、CPU、メモリ、ディスク、ネットワーク、電源状態、ゲスト OS) とパフォーマンスメトリクス (CPU 使用率、メモリ使用率、ディスク IOPS、スループット) を収集するために必要なすべてのアクセスを提供します。

推奨事項: ルートデータセンターレベルで組み込みの読み取り専用ロールを持つ専用 vCenter ユーザーを作成します。

Hyper-V ホスト (Windows)

検出ツールは、WinRM (HTTP の場合はポート 5985、HTTPS の場合はポート 5986) 経由で Hyper-V ホストに接続し、PowerShell コマンドを実行して VM インベントリ、ホストメタデータ、ストレージパフォーマンスを収集します。

各 Hyper-V ホストで最低限必要なグループメンバーシップ:

Windows グループ	必要な理由
リモート管理ユーザー	ベースライン WinRM リモートアクセス
Hyper-V 管理者	VM インベントリ収集 (VM リスト、ディスク、ネットワーク、メモリ、ゲスト OS)
Performance Monitor ユーザー	ストレージ I/O パフォーマンスカウンター (VM あたりの読み取り/書き込み IOPS とスループット)

アカウントには、ホスト OS バージョンとハードウェア UUID `root\cimv2` の名前空間への WMI 読み取りアクセスも必要です。このアクセスは、デフォルトでローカル管理者に付与されますが、管理者以外のアカウントには明示的に設定する必要があります。

推奨事項: 専用のドメインサービスアカウントを作成し、各 Hyper-V ホストのリモート管理ユーザー、Hyper-V 管理者、および Performance Monitor ユーザーグループに追加します。ドメイン管理者アカウントの使用は避けてください。

Linux サーバー (SSH)

検出ツールは、SSH (ポート 22) 経由で Linux サーバーに接続し、OS メトリクス、ネットワーク接続、サーバーインベントリを収集します。ほとんどのデータ収集は、通常のコマンドとして実行されます。少数のコマンドが試行 `sudo` され、`sudo` が使用できない場合は自動的にフォールバックされます。

最低限必要なアクセス: ターゲットサーバーにログインできる SSH ユーザーアカウント。

推奨されるアクセス: 完全なデータ収集のためにパスワードレス `sudo` で設定された同じ SSH ユーザー。

`sudo` が必要なこととそれが無いとどうなるか:

収集されるデータ	Sudo は必要ですか？	sudo なしで失ったもの
サーバー名、OS、CPU、メモリ、IP、ディスク数	いいえ	なし — 通常のユーザーとして収集
CPU、メモリ、ネットワーク使用率	いいえ	なし — 通常のユーザーとして収集
ディスク IOPS、スループット、スペース	いいえ	なし — 通常のユーザーとして収集
ネットワークインターフェイスの設定	いいえ	なし — 通常のユーザーとして収集
実行中のプロセス (名前、PID、コマンド)	いいえ	なし — 通常のユーザーとして収集
サーバー UUID と SMBIOS UUID	はい	エクスポートで UUID フィールドが空です
ハードウェアメーカー (物理検出と仮想検出)	はい	リソースタイプ検出は古いディストリビューションでは精度が低い
LVM 論理ボリューム検出	はい	LVM ボリュームが検出されません。ボリュームタイプが「不明」と表示される場合があります
プロセス名と PID を使用したネットワーク接続	はい	接続は引き続き収集されますが、プロセス属性はありません (PID 列とプロセス名列は空です)
Oracle Database OS レベルの検出 (Oracle 認証情報が設定されていない場合のフォールバック)	はい	Oracle の基本的なプレゼンス検出 (oratab、プロセスモニター) は sudo なしで機能します。詳細なコレクション (opatch、lsnrctl、ASM、Grid Infrastructure) には、Oracle ホームバイナリへのパスワードレス <code>sudo -u oracle</code> アクセスが必要です。

Note

`sudo` がないために検出ツールが完全に失敗することはありません。何ができるかを収集し、部分的な結果をレポートします。ただし、最も完全なデータ、特にプロセス名を使用したネットワーク依存関係マッピングについては、パスワードレス `sudo` をお勧めします。

ターゲットサーバーで必要なユーティリティ: 次のコマンドを使用できます (ほとんどの Linux ディストリビューションにデフォルトでインストールされています)。`ss` または、`netstat`、`lsblk`、`topps`、`free`、`ip`、`dfhostname`、`catnproc`、および `grep`。追加データのオプション: `iostat` (ディスク I/O 詳細)、`dmidecode` (ハードウェア UUID)、`smartctl` (ディスクインターフェイスタイプ)、`lvdisplay` (LVM 検出)。

Windows サーバー (WinRM) — OS メトリクス

検出ツールは WinRM 経由で Windows サーバーに接続し、OS メトリクス (サーバーのインベントリ、パフォーマンス、ストレージ、ネットワークインターフェイス、実行中のプロセス) を収集します。

最低限必要なアクセス: ターゲットサーバーへのリモートアクセスを持つ WinRM 対応ユーザーアカウント。

推奨されるアクセス: 完全なデータ収集のためにローカル管理者グループのユーザー。

アクセス許可レベル	有効にする内容	それなしで失ったもの
リモート管理ユーザー (WinRM アクセス)	すべての Windows データ収集	サーバーから収集されるデータがない
への WMI 読み取りアクセス <code>root\cimv2</code>	サーバー名、OS バージョン、メモリ、UUID、BIOS シリアル、ディスク容量	サーバーインベントリフィールドが空です
Performance Monitor ユーザー	CPU 使用率、ネットワークスループット、ディスク IOPS、スループット	収集されていないパフォーマンスメトリクス
ローカル管理者	プロセス所有者名の実行	プロセスリストが収集されましたが、ユーザー/所有者列が空です

Note

ローカル管理者メンバーシップは、暗黙的に Performance Monitor ユーザーにアクセス権と WMI 読み取りアクセス権を付与するため、上記のすべての要件を満たします。

Note

Windows での Oracle Database OS レベルのフォールバック検出は、Oracle 認証情報が設定されていない場合、レジストリクエリと コマンド `Get-Service` および `Get-Process` コマンドを使用します。これらのコマンドには、標準のリモート管理ユーザーアクセスのみが必要です。OS メトリクスの要件を超える追加のアクセス許可は必要ありません。

Windows サーバー (WinRM) — SQL Server コレクション

SQL Server コレクションは、Windows サーバー上の SQL Server インスタンス、レポートサービス (SSRS)、統合サービス (SSIS) を検出します。

推奨されるアクセス: 各ターゲット Windows サーバーのローカル管理者グループのユーザー。

SQL Server 検出は複数の WMI 名前空間をクエリし、一部のオペレーションで昇格されたアクセスを必要とするため、ローカル管理者をお勧めします。

検出された内容	必要なアクセス許可	それなしで失ったもの
SQL Server データベースエンジンインスタンス (バージョン、エディション、ステータス)	root\Microsoft\Sql Server\ComputerManagement* 名前空間への WMI 読み取りアクセス	SQL Server インスタンスが検出されない
SQL Server Reporting Services (SSRS)	root\Microsoft\Sql Server\ReportServer 名前空間への WMI 読み取りアクセス	SSRS コンポーネントが検出されない
SSRS URL とポートの設定	昇格された権限 (ローカル管理者)	SSRS URL 予約の詳細がありません

検出された内容	必要なアクセス許可	それなしで失ったもの
SQL Server Integration Services (SSIS)	レジストリ読み取りアクセス (HKLM)	SSIS バージョンとエディションがありません
Port-to-service関連付け	TCP リスナー列挙へのアクセス	リッスンポートを SQL Server サービスに関連付けることはできません

Note

SQL Server コレクションは Windows 専用です。検出ツールは、SQL Server 検出用の Linux サーバーをスキップします。

Oracle データベース (SQL)

Oracle Database コレクションは、ポート 1521 (またはカスタムポート) の Oracle インスタンスに直接接続して、データベースメタデータを収集します。検出ツールには DBA または SYSDBA 権限は必要ありません。

最低限必要なアクセス: SELECT_CATALOG_ROLE 許可を持つ読み取り専用の Oracle サービスアカウント。

検出ツールは Diagnostics Pack または Tuning Pack ビューにアクセスしないため、データ収集に追加の Oracle ライセンスは必要ありません。

アカウントを作成する SQL:

```
CREATE USER discovery_user IDENTIFIED BY <password>;
GRANT CREATE SESSION TO discovery_user;
GRANT SELECT_CATALOG_ROLE TO discovery_user;
```

OS レベルのフォールバック: データベース認証情報が設定されていないか、接続が失敗した場合、検出ツールは既存の SSH または WinRM OS 認証情報を使用して Oracle のインストールを検出します。OS メトリクスモジュールが必要とする以上の追加の Oracle 固有の OS アクセス許可は必要ありません。

ネットワークコレクション

ネットワークコレクションは、サーバーのオペレーティングシステムに応じて異なるプロトコルを使用します。

サーバー OS	プロトコル	必要となる許可
Linux	SSH	接続データの通常のユーザー。プロセスレベルの詳細 (PID とプロセス名) にはパスワードレス sudo をお勧めします。
Linux	SNMPv2	TCP MIB (tcpConnState、tcpConnectionProcess) およびホストリソース MIB (hrSWRunName) にアクセスできる読み取り専用コミュニティ文字列。
Linux	SNMPv3	SNMPv2 と同じ MIBs への読み取りアクセス権を持つ USM ユーザー。noAuthNoPriv、authNoPriv、および authPriv セキュリティレベルをサポートします。
Windows	WinRM	root\StandardCIMV2 名前空間への WMI 読み取りアクセス (MSFT_NetTCPConnection クラス)。

クイックリファレンス: ユースケース別の最小アクセス許可

ユースケース	アカウントの種類	最小アクセス許可
VMware VM 検出	vCenter ユーザー	ルートデータセンターレベルでの読み取り専用ロール
Hyper-V VM 検出	Windows ドメインまたはローカルアカウント	リモート管理ユーザー + Hyper-V 管理者 + 各ホストの Performance Monitor ユーザー
Linux OS メトリクス — フルデータ	SSH ユーザー	パスワードレス sudo

ユースケース	アカウントの種類	最小アクセス許可
Linux OS メトリクス — 部分データ	SSH ユーザー	通常のユーザー (sudo なし)。UUID、製造元、LVM、ネットワークプロセス情報がありません。
Windows OS メトリクス — フルデータ	WinRM ユーザー	ローカル管理者
Windows OS メトリクス — 基本データ	WinRM ユーザー	リモート管理ユーザー + への WMI 読み取りアクセス root\cimv2
SQL Server データベース検出	WinRM ユーザー	ローカル管理者
Oracle データベース検出 (SQL)	Oracle サービスアカウント	SELECT_CATALOG_ROLE (読み取り専用、DBA または SYSDBA なし)
ネットワークコレクション — Linux (SSH)	SSH ユーザー	プロセスレベルのデータにはパスワードレス sudo、接続データのみには通常のユーザー
ネットワークコレクション — Linux (SNMP)	SNMP コミュニティ文字列または USM ユーザー	TCP およびホストリソース MIBs への読み取りアクセス
ネットワークコレクション — Windows	WinRM ユーザー	への WMI 読み取りアクセス root\StandardCIMV2

データ収集

検出ツールの収集スケジュール

最初の検出コレクション後、リソースの競合を避けるために、検出ツールはずらされたスケジュールで引き続き実行されます。

- VMware 検出 – 1 時間ごと (:00 UTC)
- Hyper-V 検出 – 1 時間ごと (:20 UTC)

検出ツールは、以下の独立したモジュールを通じて OS メトリクスも収集します。各モジュールには独自のずらしスケジュールがあります。

- ネットワークメトリクス – 15 秒ごとに、大規模な環境では頻度が低下する可能性があります
- サーバーパフォーマンスメトリクス – 10 分ごと (:03、:13、:23、:33、:43、:53 UTC)
- ストレージパフォーマンスメトリクス – 10 分ごと (:07、:17、:27、:37、:47、:57 UTC)
- 実行中のプロセス – 時間単位 (:40 UTC)
- サーバープロビジョニングデータ – 毎日 (00:05 UTC)
- ストレージプロビジョニングデータ – 毎日 (00:35 UTC)
- ネットワークインターフェイス – 毎日 (01:05 UTC)
- SQL Server 検出 – 毎日 (03:05 UTC)
- Oracle データベース検出 – 毎日 (05:05 UTC)

Collect data now を使用して、各 OS メトリクスモジュールを個別に起動、停止、またはトリガーできます。

コレクションを手動で実行するには、アクションメニューから以下を選択します。

- 開始 — 検出モジュールを有効にします。
- Stop – 検出モジュールを無効にします。
- 今すぐデータを収集する — 検出をすぐに開始します。このオプションは、ネットワークに変更を加えた後などに使用します。

これらのアクションはモジュールごとに適用されます。OS メトリクスモジュールは個別に制御できます。

OS データ収集の試行

新しいサーバーが検出されると、検出ツールは各 IP アドレスとホスト名に対して設定された各認証情報を試行します。検出ツールは有効な認証情報を検出した後、新しい認証情報を追加しない限り、その認証情報を引き続き使用します。

収集に失敗すると、検出ツールは 3 分、30 分、2 時間、6 時間後にサーバーのネットワークデータの収集を試みます。4 回試行に失敗すると、検出ツールは 6 時間に 1 回、設定されたすべての認証情報を引き続き試します。

検出されたインベントリ

検出ソースを設定すると、検出ツールのステータスフレームで検出されたサーバーの数の値が増加し始めます。設定されたソースの検出ステータスは、コレクションモジュールフレームで Enabled に変わります。インベントリページには、VMware VMsHyper-V VMs、インポートされたサーバーなど、設定されたすべてのソースのサーバーが表示されます。各サーバーは、モジュールごとにソースとコレクションのステータスを表示します。

検出されたインベントリページに移動して、検出ツールが検出したサーバーを確認します。このページから、インベントリのダウンロードを選択して、すべての設定済みソースの MPA ファイル、パフォーマンス使用率データ、データベース情報、server-to-server通信情報など、最大 30 日間の収集データを含む ZIP ファイル (discovery_tool_export.zip) をダウンロードします。

検出ツールが引き続き動作している間に ZIP ファイルをダウンロードし、部分的な結果を取得できます。このファイルを[移行評価](#)にアップロードして、移行のビジネスケースを取得します。

エクスポートオプション

データをエクスポートするときは、次のオプションを使用してエクスポートをカスタマイズできます。

日付範囲

開始日と終了日を選択して、その期間内に収集されたデータのみをエクスポートします。両方の日付が含まれます。最大日付範囲は 30 日です。

Note

検出ツールは、収集されたデータを最大 30 日間保存します。30 日以上が必要な場合は、30 日ごとに増分エクスポートを実行してすべてのデータをキャプチャします。

モジュールの選択

エクスポートに含めるデータモジュールを選択します。すべてのモジュールをエクスポートすることも、特定のモジュールを選択することもできます。

モジュール	説明
VMware データ	vCenter サーバーからの仮想マシンインベントリ

モジュール	説明
Hyper-V データ	Hyper-V ホストからの仮想マシンインベントリ
ネットワークデータ	サーバー間のネットワーク接続
サーバーインベントリ	サーバーハードウェアと OS 情報
サーバーパフォーマンスメトリクス	CPU、メモリ、ネットワーク使用率
サーバーストレージのパフォーマンス	ディスク IOPS とスループット
ストレージ設定	ディスクとボリュームの設定
ネットワークインターフェイス	ネットワークアダプタの詳細
プロセスメトリクス	実行中のプロセス
SQL Server データ	SQL Server データベースインベントリ
Oracle Database データ	Oracle データベースインベントリ – CDBs、PDBs、機能、オプション、コンポーネント

モジュールを選択しない場合、検出ツールは使用可能なすべてのデータをエクスポートします。

収集されたデータポイント

検出ツールは、VMware、Hyper-V、OS メトリクス、データベース、ネットワークコンポーネント全体の包括的なデータを収集します。以下のセクションでは、各コンポーネント用に収集された特定のデータポイントについて詳しく説明します。

VMware データ収集

この表は、検出ツールによって収集された VMware 仮想マシン情報を示しています。

名前	型	Category	サンプル値
vm_name	String	VM 情報	「w2k22-snmpd-v2-en-us-mssql-2022-testcase4-1」
vm_id	String	VM 情報	「vm-30920」
vm_uuid	String	VM 情報	「4201ecf8-cc44-ee7e-01da-34dfb2acf6c0」
powerstate	String	VM 情報	「poweredOn」
ホスト	String	VM 情報	「esxi-70-node1.testlab.local」
primary_ip_address	String	VM 情報	「192.168.0.52」
cpus	整数	VM 情報	2
メモリ	整数	VM 情報	4096
total_disk_capacity_mib	整数	VM 情報	32768
os_according_to_the_configuration_file	String	VM 情報	「Microsoft Windows Server 2016 以降 (64 ビット)」
max_cpu_usage_pct_dec	浮動小数点数	VM のパフォーマンス	79.33
avg_cpu_usage_pct_dec	浮動小数点数	VM のパフォーマンス	45.06
max_ram_usage_pct_dec	浮動小数点数	VM のパフォーマンス	63.99
avg_ram_util_pct_dec	浮動小数点数	VM のパフォーマンス	29.27

Hyper-V データ収集

この表は、検出ツールによって収集された Hyper-V 仮想マシン情報を示しています。

名前	型	Category	サンプル値
vm_name	String	VM 情報	「win2022-hyperv-test-01」
vm_id	String	VM 情報	「a1b2c3d4-e5f6-7890-abcd-ef1234567890」
powerstate	String	VM 情報	「実行中」
cpus	整数	VM 情報	4
memory_mb	整数	VM 情報	8192
disk_paths	String	ディスク	「C:\VMs\disk1.vhdx」
disk_size_gb	浮動小数点数	ディスク	127.0
network_adapters	String	Network	「00:15:5D:01:02:03」
ip_addresses	String	Network	「10.0.1.50」
host_name	String	ホスト	「hyperv-host-01.example.com」
host_os_version	String	ホスト	「Windows Server 2022 Datacenter」
cluster_name	String	ホスト	「FailoverCluster01」
ハイパーバイザー	String	VM 情報	「ハイパー V」

インポートされたサーバーデータ

インポートされたサーバーは自動検出されません。これらは CSV ファイルを介してインポートされます。検出ツールは、インポートされたサーバーのハイパーバイザーレベルのデータを収集しません。代わりに、インポート時に各サーバーに関連付けられた OS 認証情報を使用して、データベース、ネットワーク、OS メトリクスデータを収集します。

検出ツールの OS 関連データ

検出ツールは、SSH (Linux) と WinRM (Windows) を介してサーバーインベントリ、パフォーマンス、ストレージ、ネットワークインターフェイスを収集し、データを処理します。次の表は、収集されたデータポイントを示しています。

サーバーインベントリ (server_inventory.csv)

サーバーのプロビジョニング (ハードウェアと OS の設定) と集約されたストレージパフォーマンスを組み合わせます。24 時間ごとに収集されます。

名前	型	Category	サンプル値
server_id	String	Server Info	「vm-web-server-01」
server_name	String	Server Info	「web-server-01」
resource_type	String	Server Info	「virtual_machine」
power_state	String	Server Info	「実行中」
os_type	String	Server Info	「Linux」
os_name	String	Server Info	「Amazon Linux」
os_version	String	Server Info	「2023」
primary_hostname	String	Server Info	「web-server-01.example.com」
primary_ip_address	String	Server Info	「10.0.2.101」
ネットマスク	String	Server Info	「255.255.255.0」
total_num_network_cards	整数	Server Info	2
total_num_disks	整数	Server Info	1
cpu_count	整数	Server Info	4
total_memory_gb	浮動小数点数	Server Info	15.88

名前	型	Category	サンプル値
server_uuid	String	Server Info	「4201ecf8-cc44-ee7e-01da-34dfb2acf6c0」
smbios_uuid	String	Server Info	「4201ecf8-cc44-ee7e-01da-34dfb2acf6c0」
cluster_name	String	Server Info	「production-cluster-01」
hypervisor_object_id	String	Server Info	「vm-30920」
hypervisor_type	String	Server Info	「VMware」
hypervisor_version	String	Server Info	「8.0.0」
hypervisor_hostname	String	Server Info	「esxi-node1.example.com」
hypervisor_host_id	String	Server Info	「host-1234」
hypervisor_id	String	Server Info	「4201ecf8-cc44-ee7e-01da-34dfb2acf6c0」
disk_read_iops_avg	浮動小数点数	ストレージパフォーマンス	12.5
disk_read_iops_peak	浮動小数点数	ストレージパフォーマンス	245.0
disk_write_iops_avg	浮動小数点数	ストレージパフォーマンス	8.3
disk_write_iops_peak	浮動小数点数	ストレージパフォーマンス	180.0
disk_total_iops_avg	浮動小数点数	ストレージパフォーマンス	20.8
disk_total_iops_peak	浮動小数点数	ストレージパフォーマンス	425.0

名前	型	Category	サンプル値
disk_read_throughput_avg_mbps	浮動小数点数	ストレージパフォーマンス	1.2
disk_read_throughput_peak_mbps	浮動小数点数	ストレージパフォーマンス	24.5
disk_write_throughput_avg_mbps	浮動小数点数	ストレージパフォーマンス	0.8
disk_write_throughput_peak_mbps	浮動小数点数	ストレージパフォーマンス	18.0
disk_total_throughput_avg_mbps	浮動小数点数	ストレージパフォーマンス	2.0
disk_total_throughput_peak_mbps	浮動小数点数	ストレージパフォーマンス	42.5

サーバーパフォーマンスメトリクス (server_performance_metrics.csv)

CPU、メモリ、ネットワークスループットの使用率。10 分ごとにサンプリングされ、30 日間にわたって集計されます。

名前	型	Category	サンプル値
server_id	String	Server Info	「vm-web-server-01」
data_source	String	Server Info	「OS」
cpu_utilization_avg_pct	浮動小数点数	CPU	45.06
cpu_utilization_peak_pct	浮動小数点数	CPU	79.33
cpu_count	整数	CPU	4
memory_total_gb	浮動小数点数	メモリ	15.88

名前	型	Category	サンプル値
memory_utilization_avg_pct	浮動小数点数	メモリ	29.27
memory_utilization_peak_pct	浮動小数点数	メモリ	63.99
network_in_avg_mbps	浮動小数点数	Network	0.52
network_in_peak_mbps	浮動小数点数	Network	12.3
network_out_avg_mbps	浮動小数点数	Network	0.31
network_out_peak_mbps	浮動小数点数	Network	8.7
network_total_avg_mbps	浮動小数点数	Network	0.83
network_total_peak_mbps	浮動小数点数	Network	21.0

ストレージパフォーマンス (server_storage_performance.csv)

ボリュームごとのディスク I/O とスペース使用率。10 分ごとにサンプリングされ、30 日間にわたって集計されます。

名前	型	Category	サンプル値
server_id	String	Server Info	「vm-web-server-01」
data_source	String	Server Info	「OS」
disk_volume_id	String	ボリューム情報	「/dev/nvme0n1p1」
disk_mount_point	String	ボリューム情報	"/"
file_system	String	ボリューム情報	「xfs」
disk_total_gb	浮動小数点数	ディスク容量	30.0

名前	型	Category	サンプル値
disk_used_gb	浮動小数点数	ディスク容量	12.5
disk_free_gb	浮動小数点数	ディスク容量	17.5
disk_read_iops_avg	浮動小数点数	ディスク I/O	12.5
disk_read_iops_peak	浮動小数点数	ディスク I/O	245.0
disk_write_iops_avg	浮動小数点数	ディスク I/O	8.3
disk_write_iops_peak	浮動小数点数	ディスク I/O	180.0
disk_total_iops_avg	浮動小数点数	ディスク I/O	20.8
disk_total_iops_peak	浮動小数点数	ディスク I/O	425.0
disk_read_throughput_avg_mbps	浮動小数点数	ディスクスループット	1.2
disk_read_throughput_peak_mbps	浮動小数点数	ディスクスループット	24.5
disk_write_throughput_avg_mbps	浮動小数点数	ディスクスループット	0.8
disk_write_throughput_peak_mbps	浮動小数点数	ディスクスループット	18.0
disk_total_throughput_avg_mbps	浮動小数点数	ディスクスループット	2.0
disk_total_throughput_peak_mbps	浮動小数点数	ディスクスループット	42.5

ストレージ設定 (storage_config.csv)

物理ディスクハードウェアの詳細。24 時間ごとに収集されます。

名前	型	Category	サンプル値
server_id	String	Server Info	「vm-web-server-01」
disk_controller_id	String	ディスク情報	「/dev/sda」
vmdk_vhd_file_name	String	ディスク情報	「web-server-01.vmdk」
disk_volume_type	String	ディスク情報	「仮想」
disk_provisioned_gb	浮動小数点数	ディスク情報	30.0
disk_device_type	String	ディスク情報	「SCSI HDD」
disk_interface_type	String	ディスク情報	「SCSI」
disk_protocol	String	ディスク情報	「LSI ロジック SAS」

ネットワークインターフェイス (network_interfaces.csv)

ネットワークアダプタの設定。24 時間ごとに収集されます。

名前	型	Category	サンプル値
server_id	String	Server Info	「vm-web-server-01」
interface_name	String	インターフェイス情報	「eth0」
interface_index	整数	インターフェイス情報	2
mac_address	String	インターフェイス情報	「0A:1B:2C:3D:4E:5F」
adapter_type	String	インターフェイス情報	「vmxnet3」

名前	型	Category	サンプル値
virtual_network_name	String	インターフェイス情報	「VM ネットワーク」
virtual_network_id	String	インターフェイス情報	「dvportgroup-1234」
virtual_switch	String	インターフェイス情報	「vSwitch0」
ipv4_address	String	IP 設定	「10.0.2.101」
ipv4_subnet_mask	String	IP 設定	「255.255.255.0」
ipv4_gateway	String	IP 設定	「10.0.2.1」
ipv6_address	String	IP 設定	「fe80::a1b:2cff:fe3d:4e5f」
ipv6_prefix_length	整数	IP 設定	64
ipv6_gateway	String	IP 設定	「fe80::1」
dns_servers	String	IP 設定	「10.0.0.2」
dhcp_enabled	ブール値	IP 設定	false
interface_status	String	インターフェイス情報	「アップ」
vlan_id	整数	インターフェイス情報	100
is_primary	ブール値	インターフェイス情報	true

実行中のプロセス (process_metrics.csv)

実行中のプロセスのスナップショット。1 時間ごとに収集され、30 日間にわたって重複排除されます。

名前	型	Category	サンプル値
server_id	String	Server Info	「vm-web-server-01」
process_name	String	プロセス情報	「sshd」
process_id	整数	プロセス情報	1234
process_command_line	String	プロセス情報	「/usr/sbin/sshd -D」
process_user	String	プロセス情報	「ルート」

ネットワークコレクション

ネットワークコレクションモジュールは、オンプレミスデータセンター内のサーバー間の依存関係を検出するのに役立ちます。このネットワークデータは、アプリケーションがサーバー間で通信する方法を可視化することで、移行計画を加速します。

このモジュールは、VMware、Hyper-V、インポートされたサーバーなど、設定されたすべてのソースからサーバーのネットワークデータを収集します。WinRM を使用して Windows サーバーからデータを収集し、SSH、SNMPv2、SNMPv3 を使用して Linux サーバーからデータを収集します。

ネットワークデータ収集

ネットワークコレクションモジュールは、検出されたインベントリ内のサーバー間の ESTABLISHED または TIME_WAIT 状態の TCP IPv4 接続をキャプチャします。ソース IP アドレスとターゲット IP アドレスの両方が、検出ツールが検出したサーバーまたはインポートしたサーバーに属している場合にのみ、接続が出力に表示されます。外部サービス、クラウドエンドポイント、検出ツールにまだ追加されていないサーバーなど、インベントリ外の IP アドレスとの間の接続は含まれません。

この設計では、環境内のserver-to-server依存関係にネットワークデータに焦点を当てています。これは、アプリケーションの依存関係マッピングと移行ウェーブプランニングに必要な情報です。

これらのデータポイントは、接続ごとに収集されます。

名前	型	Category	サンプル値
[Source IP] (送信元 IP)	String	接続	「192.168.1.10」

名前	型	Category	サンプル値
ソースポート	整数	接続	49152
ソースプロセス ID	整数	プロセス	1234
ソースプロセス名	String	プロセス	"java"
ターゲット IP	String	接続	「192.168.1.20」
ターゲットポート	整数	接続	5432
ターゲットプロセス ID	整数	プロセス	5678
ターゲットプロセス名	String	プロセス	「postgres」
State	String	接続	「ESTABLISHED」
トランスポートプロトコル	String	接続	「TCP」
IP バージョン	String	接続	「IPv4」
カウント	整数	接続	42

Tip

ネットワーク依存関係マップの完全性を最大化するには、ネットワークデータを確認する前に、すべての検出ソース (VMware、Hyper-V、サーバー CSV インポート) を設定し、OS 認証情報を追加します。インベントリ内のサーバーが多いほど、ネットワークモジュールがキャプチャできる接続が増えます。

プライベートアドレスネットワークコレクション

デフォルトでは、ネットワークコレクションモジュールは、両方のエンドポイントが検出されたインベントリ内のサーバーである接続のみをキャプチャします。プライベートアドレス収集を有効にして、インベントリにない RFC 1918 プライベート IP アドレス (10.0.0.0/8、172.16.0.0/12、192.168.0.0/16) との間の接続をキャプチャすることもできます。

プライベートアドレスの収集を開始するには

1. コレクター設定ページで、コレクションモジュールセクションを見つけます。
2. アプリケーション検出でネットワーク接続検出を検索します。
3. Actions ドロップダウンを開きます。
4. プライベートアドレス収集の開始 を選択します。
5. モジュールのステータスが Enabled の場合、Enabled · Private Address on に変わります。

プライベートアドレスの収集を停止するには、アクションドロップダウンを開き、プライベートアドレスの収集を停止を選択します。以前に収集されたプライベートアドレスデータは、停止後も保持されます。

プライベートアドレス接続は、完全な CSV エクスポート (ZIP ファイル内) にのみ表示され、MPA CSV ファイルには表示されません。IP アドレスが既にインベントリにあるサーバーに属している場合、この設定に関係なく、検出されたサーバー ID によって常に識別されます。

この設定は再起動後も保持されます。プライベートアドレスの収集はいつでも開始または停止できます。以前に収集されたプライベートアドレスデータは、現在の設定に関係なくエクスポートされます。

SQL Server コレクション

SQL Server コレクションモジュールは、VMware、Hyper-V、インポートされたサーバーなど、設定されたすべてのソースにわたって Windows サーバーから SQL Server 情報を収集します。モジュールは WinRM を使用して各 Windows サーバーにリモート接続し、PowerShell クエリを実行します。WMI 名前空間、レジストリ、ファイルプロパティを使用して、インストールされているすべての SQL Server サービス (コンポーネント) に関する情報を収集します。

SQL Server コンポーネントは、Windows サーバーへの SQL Server デプロイの一部としてインストールされる特定のサービスまたは機能インスタンスです。検出ツールは、データベースエンジン、分析サービス、レポートサービス、統合サービスを収集します。

SQL Server データ収集

SQL Server コレクションモジュールは、SQL Server コンポーネント情報を収集します。この表は、収集された主要なデータポイントを示しています。

名前	型	Category	サンプル値
エンジンのタイプ	String	コンポーネント	sql_server
エンジンコンポーネント	ブール値	コンポーネント	はい
ステータス	String	サービス	実行、停止、StartPending
バージョン	String	サービス	2015 年 131 月 5026 日
Edition	String	サービス	Developer Edition (64 ビット)
SQL サービス名	String	サービス	MsDtsServer130、Mssql
SQL サービスタイプ	String	サービス	SQL Server サービス、統合サービスサービス
インスタンス名	String	インスタンス	MSSQLSERVER
[Display Name] (表示名)	String	サービス	SQL Server (MSSQLSERVER2017)
開始モード	String	サービス	自動、手動、無効
Service Account Name	String	サービス	NT サービス/MsDtsServer130
クラスター化されている	ブール値	設定	いいえ

Note

フル形式には、すべてのサービスタイプが含まれます。MPA 形式には、データベースエンジンコンポーネントのみが含まれます。SQL サービスのタイプと設定によっては、すべてのフィールドを使用できるわけではありません。

Oracle Database コレクション

Oracle Database コレクションを使用すると、VMware、Hyper-V、インポートされたサーバーなど、設定されたすべてのソースで Oracle データベースインスタンスを検出できます。モジュールは次のデータを収集します。

- コンテナデータベース (CDB) とプラガブルデータベース (PDB) の列挙
- バージョンとエディション
- インストールされたオプション
- 機能の使用統計
- コンポーネントインベントリ (DBA_REGISTRY)
- データファイルのサイジング
- トポロジフラグ (RAC、Data Guard、マルチテナントアーキテクチャ)

データベース接続 (SQL)

Oracle 認証情報を設定すると、検出ツールは読み取り専用サービスアカウントを使用して Oracle データベースに直接接続します。これにより、PDB の詳細、機能の使用状況、インストールされたオプションなど、完全な SQL レベルのコレクションが提供されます。すべての CSV ファイルには完全なデータが含まれています。

OS レベルのフォールバック

データベース認証情報を設定していない場合、または接続が失敗した場合、検出ツールは SSH または WinRM を使用して Oracle のインストールを検出します。データベースにアクセスせずに Oracle のホーム、リスナー、パッチ、バージョン、エディションを検出します。OS で検出されたホストの場合、エクスポートされた CDB CSV にはインスタンス名、ホスト名、バージョン、エディションが含まれます。PDB、機能、オプション、コンポーネント CSV ファイルには行が含まれていません。

Oracle コレクションは、エクスポート ZIP に次の CSV ファイルを生成します。各テーブルには、ファイルごとに収集されたデータポイントが記述されています。

CDB データ (oracle_data_cdb_full.csv)

CDB インスタンスごとに 1 行。この表は、収集された CDB データポイントを示しています。

名前	型	Category	サンプル値	ソース
インスタンス名	String	アイデンティティ	「ORCL」	SQL、OS
ホスト名	String	アイデンティティ	「oracledb01.example.com」	SQL、OS
DB Name	String	アイデンティティ	「ORCL」	SQL のみ
DB 一意の名前	String	アイデンティティ	「ORCL_PRIMARY」	SQL のみ
DBID	整数	アイデンティティ	1234567890	SQL のみ
バージョン	String	バージョン	「19.0.0.0」	SQL、OS
バージョンフル	String	バージョン	「19.21.0.0.0」	SQL のみ
Edition	String	バージョン	「エンタープライズエディション」	SQL、OS
データベースタイプ	String	設定	「SINGLE」	SQL のみ
データベースロール	String	設定	「プライマリ」	SQL のみ
オープンモード	String	設定	「書き込みを読み取る」	SQL のみ
ログモード	String	設定	「アーカイブ」	SQL のみ
プラットフォーム	String	設定	「Linux x86 64 ビット」	SQL のみ
CDB フラグ	String	設定	「はい」	SQL のみ
は RAC です	String	トポロジ	「はい」または「いいえ」	SQL のみ

名前	型	Category	サンプル値	ソース
Data Guard のスタンバイ	String	トポロジ	「はい」または「いいえ」	SQL のみ
保護モード	String	トポロジ	「最大パフォーマンス」	SQL のみ
データガードブロッカー	String	トポロジ	「ENABLED」	SQL のみ
NLS 文字セット	String	設定	「AL32UTF8」	SQL のみ
PDB 数	整数	サイジング	3	SQL のみ
フラッシュバックオン	String	設定	「はい」	SQL のみ
検出パス	String	メタデータ	「phase2」または「phase1b」	SQL、OS

PDB データ (oracle_data_pdbs_full.csv)

PDB (プラグ可能なデータベース) ごとに 1 行。この表は、収集された PDB データポイントを示しています。

Note

PDB データには Oracle データベース認証情報 (SQL 接続) が必要です。OS レベルのフォールバックでは、この CSV は入力されません。

名前	型	Category	サンプル値	ソース
CDB インスタンス名	String	アイデンティティ	「ORCL」	SQL のみ
PDB 名	String	アイデンティティ	「APPPDB1」	SQL のみ

名前	型	Category	サンプル値	ソース
オープンモード	String	ステータス	「書き込みを読み取る」	SQL のみ
ライフサイクルステータス	String	ステータス	「NORMAL」	SQL のみ
テーブルスペース数	整数	サイジング	5	SQL のみ
データファイル数	整数	サイジング	12	SQL のみ
合計サイズバイト	整数	サイジング	5368709120	SQL のみ
ユーザスキーマの数	整数	サイジング	8	SQL のみ
DB リンク数	整数	接続	2	SQL のみ
インストールされたコンポーネント	String	設定	「APEX;JVM;XML」	SQL のみ
暗号化されたテーブルスペースの数	整数	セキュリティ	1	SQL のみ

機能使用状況データ (oracle_data_features_full.csv)

DBA_FEATURE_USAGE_STATISTICS の機能使用状況エントリごとに 1 行。この表は、収集された機能使用状況データポイントを示しています。

Note

機能使用状況データには、Oracle データベース認証情報 (SQL 接続) が必要です。OS レベルのフォールバックでは、この CSV は入力されません。

名前	型	Category	サンプル値	ソース
CDB インスタンス名	String	アイデンティティ	「ORCL」	SQL のみ

名前	型	Category	サンプル値	ソース
Name	String	機能	「パーティショニング (ユーザー)」	SQL のみ
検出された使用状況	整数	使用方法	42	SQL のみ
現在使用されている	String	使用方法	「TRUE」	SQL のみ
初回使用日	DateTime	使用方法	「2024-01-15T00:00:00」	SQL のみ
最終使用日	DateTime	使用方法	「2026-06-01T00:00:00」	SQL のみ

オプションデータ (oracle_data_options_full.csv)

CDB ごとに V\$OPTION エントリごとに 1 行。この表は、収集されたインストール済みオプションデータポイントを示しています。

Note

オプションデータには、Oracle データベース認証情報 (SQL 接続) が必要です。OS レベルのフォールバックでは、この CSV は入力されません。

名前	型	Category	サンプル値	ソース
CDB インスタンス名	String	アイデンティティ	「ORCL」	SQL のみ
オプション名	String	オプション	「高度な分析」	SQL のみ
インストール済み	String	オプション	「TRUE」	SQL のみ
コンテナ ID	整数	オプション	0	SQL のみ

コンポーネントデータ (oracle_data_components_full.csv)

DBA_REGISTRY コンポーネントごとに 1 行。この表は、収集されたコンポーネントデータポイントを示しています。

 Note

コンポーネントデータには、Oracle データベース認証情報 (SQL 接続) が必要です。OS レベルのフォールバックでは、この CSV は入力されません。

名前	型	Category	サンプル値	ソース
CDB インスタンス名	String	アイデンティティ	「ORCL」	SQL のみ
PDB 名	String	アイデンティティ	「APPPDB1」	SQL のみ
コンポーネント ID	String	コンポーネント	「APEX」	SQL のみ
コンポーネント名	String	コンポーネント	「Oracle Application Express」	SQL のみ
バージョン	String	コンポーネント	「22.1.0.15.0」	SQL のみ
ステータス	String	コンポーネント	「VALID」	SQL のみ
Schema	String	コンポーネント	「APEX_220100」	SQL のみ

セキュリティに関する考慮事項

検出ツール VM の保護

検出ツールは、すべての認証情報、ログ、顧客データを検出ツール VM に保存するため、検出ツール VM のセキュリティは重要です。

検出ツール VM にはデフォルトで ssh アクセスが無効になっており、クライアントからのみアクセスできますvCenter UI -> "Launch Web Console"。

検出ツール VM には、ユーザー「検出」のデフォルトのログインパスワード「パスワード」が付属しています。

- デプロイ後すぐにこのパスワードを更新することをお勧めします。
- vCenter を使用してログインenablenesshした後に コマンドを使用して ssh を有効にする場合は、パスワードを更新する必要がありますLaunch Web Console。このコマンドが呼び出されるたびに、リセットする必要があることに注意してください。

認証情報の保護

認証情報管理の一般的なベストプラクティス

- 認証情報を安全に保存する
- すべての認証情報を定期的に更新する
- パスワードマネージャーまたは安全なボールドを使用する
- 認証情報の使用状況をモニタリングする
- 最小特権の原則に従い、必要な最小限のアクセス許可のみを付与する

SNMP v2 認証情報

- デフォルト以外の複雑なコミュニティ文字列を使用する
- 「パブリック」や「プライベート」などの一般的な文字列を避ける
- コミュニティ文字列をパスワードとして扱う

SNMP v3 認証情報

- 認証とプライバシーの両方を有効にする
- 強力な認証プロトコルを使用する (MD5 よりも SHA が推奨)
- 強力な暗号化プロトコルを使用する (DES よりも AES が優先)
- 認証とプライバシーの両方に複雑なパスワードを使用する
- 一意のユーザー名を使用する (共通名は避ける)

WinRM 認証情報

- WinRM 証明書チェックを無効にしないでください。
- 最小限のアクセス許可で専用サービスアカウントを作成することをお勧めします。
- SQL Server コレクションが必要でない限り、ドメイン管理者アカウントまたはローカル管理者アカウントを使用しないでください。SQL Server コレクションでは、複数の WMI 名前空間をクエリし、昇格されたコマンドを使用するため、ローカル管理者アクセスが必要です。SQL Server を検出しない OS メトリクスの場合、リモート管理ユーザー、Performance Monitor ユーザー、および WMI 読み取りアクセスを持つ管理者以外のアカウントで十分root\cimv2です。モジュールごとの詳細については[the section called “必要なアクセス許可”](#)、「」を参照してください。

Hyper-V 認証情報

- Hyper-V 管理アクセス許可を最小限に抑えた専用サービスアカウントを使用します。
- ドメイン管理者アカウントの使用は避けてください。
- Hyper-V 認証情報は、NTLM (HTTPS のみ) および Kerberos 認証をサポートします。
- 検出ツールは、SQLCipher を使用して保管時に暗号化された認証情報を保存します。

Oracle 認証情報

- SELECT_CATALOG_ROLE のみを使用する専用の読み取り専用サービスアカウントを使用します。DBA、SYSDBA、または SYSOPER 権限を使用しないでください。
- 検出ツールは読み取りオペレーションのみを実行し、Oracle データベースに書き込むことはありません。
- 検出ツールは Diagnostics Pack または Tuning Pack ビューにアクセスしないため、追加の Oracle ライセンスは必要ありません。
- Oracle サービスアカウントのパスワードを定期的にローテーションし、検出ツールで認証情報を更新します。

認証情報ストレージ

検出ツールは、データベース暗号化キーを使用して保管中の保存された認証情報を暗号化します。systemd 250 以降のシステムでは、このキーは systemd-creds を使用して暗号化されます。古いシステムでは、キーはアクセス許可で保護されたファイルとして保存されます。いずれの場合も、検出ツールホストへのルートアクセス権を持つ攻撃者は暗号化キーにアクセスし、保存された認証情報を復号できます。検出ツールホストへのアクセスを制限し、環境内の特権システムとして扱います。

注意して自動接続機能を使用する

検出ツールは、OS レベルの収集にサーバーに認証情報を割り当てるために、自動接続と手動の 2 つのメカニズムを使用します。OS レベルのコレクションには、ネットワーク、SQL Server、Oracle Database、OS メトリクスモジュールが含まれます。これらのモジュールは、VMware VMs、Hyper-V VMs、インポートされたサーバーなど、すべてのソースから個々のサーバーに接続します。

手動: サーバーは、特定の認証情報に手動で関連付けることができます。この場合、検出ツールは、成功または失敗に関係なく、その認証情報のみを使用します。そのサーバーのコレクションステータスを手動でモニタリングし、調整する必要があります。

自動接続: 認証情報がサーバーに手動で関連付けられていない場合、検出ツールはそのサーバーの自動接続メカニズムを使用します。つまり、次のようになります。

- 各コレクションラウンドの開始時に、検出ツールは (OS タイプに基づいて) そのサーバーで利用できる認証情報のリストを取得し、「自動接続可能」に設定されます。
- 次に、検出ツールは、ループ内のサーバーに対してすべての認証情報をテストします。
 - 動作している認証情報が見つかった場合、サーバーのコレクションラウンドは成功です。検出ツールはそれを記憶し、次回初めて試行します。
 - 動作中の認証情報が見つからない場合、サーバーのコレクションラウンドは失敗しています。
 - ネットワークモジュール: サーバーはバックオフスケジュールを使用し、各障害から 3 分、30 分、2 時間、6 時間後に次のコレクションラウンドを開始します (指数バックオフに似ています)。
 - SQL Server コレクション: 検出ツールは再試行しません。サーバーごとに毎日 1 回試行されます。

影響/リスク:

自動接続は、システムでリスクが確実に軽減される場合にのみ使用してください。

リスク 1: 複数の間違った認証情報で自動接続を設定した場合、サーバーに対して自動的に試行すると、ロックアウトポリシーが設定された本番環境でアカウントロックアウトがトリガーされる可能性があります。たとえば、データセンターは、SSH ログインの試行が 3 回失敗した後に VMs をロックダウンするように設定できます。この場合、3 つの間違った SSH 認証情報に対して自動接続が設定されている場合、正当なアカウントロックアウトが発生します。ロックアウトが複数のシステムで発生すると、重要なビジネスプロセスが影響を受け、依存システムにカスケード障害が発生する可能性があります。さらに、セキュリティオペレーションセンターでは、大量の認証失敗イベントによるアラートストームが発生し、偽陽性のセキュリティインシデントが発生してリソースが枯渇し、実際の攻撃がマスクされる可能性があります。

リスク 2: 検出ツール (Knows Discovery Tool パスワード) にアクセスできるアクターは、多数のテスト認証情報を設定し、自動接続を使用して成功した認証情報を見つけることで、すべてのサーバーで OS 認証情報をブルートフォースできます。

緩和策:

次のガイドラインに従ってください:

- 検出ツールのパスワードが適切に保護され、承認されたアクターのみが知っていることを確認します。
- ロックアウトポリシーが設定されている場合は、環境に適切な認証情報が入力されていることを確認します。個々の VMs の運用負荷を最小限に抑えるために、アカウントロックアウトポリシーがない場合でも、既知の有効な認証情報を設定することをお勧めします。
- 「自動接続」はオプトイン機能です。アカウントロックアウトが環境にとって懸念される場合は、選択せず、手動で認証情報を割り当てる必要があります。

CSV インポートのセキュリティ

CSV ファイルを使用してサーバーをインポートする場合は、次のセキュリティへの影響を考慮してください。

- CSV ファイルには、内部サーバーのホスト名または IP アドレスが含まれている場合があります。機密データとして扱います。
- 列 `os_credential_name` と `oracle_credential_name` 列は、事前設定された認証情報をわかりやすい名前参照します。CSV にはシークレットは含まれません。
- All-or-nothing 検証: CSV のいずれかの行が無効である場合、アップロード全体が拒否されます。これにより、混乱した状態を引き起こす可能性のある部分的なインポートを防ぐことができます。

- インポートされたサーバーはインベントリにすぐに表示され、収集の対象となります。インポートする前に、OS と Oracle の認証情報の範囲が正しく設定されていることを確認してください。

アクセスの取り消しに関する考慮事項

アクセスを取り消すと、削除の範囲は特定のソースに限定されます。

- vCenter アクセスを取り消すと、vCenter データのみが削除されます。Hyper-V またはインポートされたサーバーデータには影響しません。
- Hyper-V アクセスを取り消すと、Hyper-V データのみが削除されます。VMware またはインポートされたサーバーデータには影響しません。
- インポートされたサーバーを削除するとインベントリから削除されますが、ダウンストリームのコレクションデータ (ネットワーク、データベース、OS メトリクス) は保持されます。
- ソース固有のインベントリデータをすべて削除するには、各ソースを個別に取り消しまたは削除する必要があります。ダウンストリーム収集データ (ネットワーク、データベース、OS メトリクス) は、すべてのソースが取り消された後も保持されます。

トラブルシューティング

vCenter への検出ツールの接続の検証

VMware モジュールの設定エラーが発生した場合は、以下の手順に従って接続を確認します。

検出ツール VM にアクセスする

- 検出ツール VM にログインし、vCenter でリモートコンソールを開きます。
 - ユーザー名: discovery
 - パスワード: password

vCenter 接続のテスト

1. vCenter API アクセスをテストする:

```
curl -v --insecure -u <username>:<password> https://<vcenter-ip-or-hostname>:443/mob
```

2. 期待される成功出力:

```
[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
```

SSL 証明書をテストする

1. 次のコマンドを実行します。

```
openssl s_client -showcerts -servername <hostname> -connect <hostname>:443
```

2. 期待される成功出力:

- vSphere 証明書の詳細を表示する必要があります
- ポート 443 での SSL/TLS 接続の検証

```
[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  
---
```

WinRM のトラブルシューティング

WinRM との接続に問題がある場合は、以下の手順に従って接続をテストします。これらの手順は、検出ツールが WinRM を使用して Hyper-V ホストと通信するため、Hyper-V 接続の問題にも適用されます。

ポート 5985 (HTTP) と 5986 (HTTPS) を使用して基本的な WinRM 接続をテストします。ポート 5986 (HTTPS) で接続が機能することを確認する必要があります

```
# 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
```

上記のテストが失敗した場合は、証明書の検証を無効にして PowerShell セッションを確立してみてください。

```
$cred = Get-Credential  
$so = New-PsSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck  
Enter-PSsession -ComputerName <HOST> -Credential $cred -Port 5985 -SessionOption $so
```

Kerberos のトラブルシューティング

Windows サーバーからデータを収集するときに Kerberos 認証に障害が発生した場合は、次のセッションを使用して一般的な問題を診断し、解決します。

ネットワーク要件の検証

Kerberos 認証のトラブルシューティングを行う前に、検出ツールが必要なネットワークエンドポイントに到達できることを確認します。

Kerberos のネットワーク要件を確認するには

1. ドメインコントローラーへの DNS 解決を確認します。検出ツール VM から次のコマンドを実行します。

```
nslookup dc01.example.com
```

または、dig を使用することもできます。

```
dig dc01.example.com
```

DNS 解決に失敗した場合は、Active Directory ドメインを解決できる DNS サーバーを使用するように検出ツール VM が設定されていることを確認します。ネームサーバーエントリがドメイン DNS サーバーを指し/etc/resolv.confでいることを確認します。

2. ポート 88 の Key Distribution Center (KDC) への接続を確認します。次のコマンドを実行します。

```
nc -zv dc01.example.com 88
```

正常な出力:

```
Connection to dc01.example.com 88 port [tcp/kerberos] succeeded!
```

接続が失敗した場合、ファイアウォールルールが検出ツール VM からポート 88 のドメインコントローラーへのトラフィックをブロックしていないことを確認します。

3. WinRM ポート上のターゲット Windows サーバーへの接続を確認します。以下の コマンドを実行します。

```
nc -zv <windows-server> 5985  
nc -zv <windows-server> 5986
```


接続が失敗した場合、ターゲットサーバーで WinRM が有効になっていること、およびファイアウォールルールでポート 5985 および 5986 でのインバウンドトラフィックが許可されていることを確認します。

一般的な Kerberos の問題

以下は、一般的な Kerberos の問題とその解決策です。

大文字と小文字の区別エラー

症状: 認証中に「サーバーが Kerberos データベースに見つかりません」というエラーが表示されます。

このエラーは通常、Kerberos 領域名が大文字でない場合に発生します。Kerberos レalm は、`krb5.conf` ファイルの大文字で指定する必要があります。例えば、`example.com` ではなく `EXAMPLE.COM` を使用します。

アカウントロックアウトの防止

検出ツールはバックオフメカニズムを使用して、アカウントのロックアウトが認証試行に失敗するのを防ぎます。サービスアカウントがロックアウトされた場合は、 の検出ツールウェブ UI を使用してコレクションモジュールを停止して開始することで、コレクションプロセスをリセットできます `https://<discovery-tool-vm-ip>:5000`。

CLI から kinit が失敗する

次の表に、一般的な kinit エラーとその解決策を示します。

エラー	原因	ソリューション
領域の KDC が見つかりません	KDC ホスト名または IP アドレスに到達できないか、領域がで設定されていません <code>krb5.conf</code> 。	<code>krb5.conf</code> ファイルに正しい KDC ホスト名と領域が含まれていることを確認します。ポート 88 で DNS 解決と KDC へのネットワーク接続を確認します。
事前認証に失敗しました	サービスアカウントのパスワードが正しくありません。	パスワードを確認して、もう一度試してください。アカウントがロックアウトされている場合は、再試行する前に

エラー	原因	ソリューション
		Active Directory でロックを解除します。
Kerberos データベースにクライアントが見つかりません	プリンシパル名が Active Directory のどのアカウントとも一致しません。	プリンシパル名が、大文字と小文字を含むアカウント名と正確に一致することを確認します。レルムを大文字username@REALM にした 形式を使用します。
KDC のネットワークアドレスを解決できません	DNS は KDC ホスト名を解決できません。	で DNS 設定を確認します/etc/resolv.conf 。DNS サーバーが KDC ホスト名を解決できることを確認します。nslookup または を使用してテストしますdig。

kinit が成功してもコレクションが失敗する

がkinit成功してもデータ収集が失敗する場合は、以下を確認してください。

1. コレクションに使用されるプリンシパル名が、中に使用されたケースとkinit正確に一致することを確認します。
2. サービスアカウントにターゲットサーバーに必要なアクセス許可があることを確認します。
3. ターゲットサーバーで WinRM が有効になっていることを確認します。
4. コレクションに使用されるホスト名が Active Directory に登録されているホスト名と一致することを確認します。

Kerberos は一部のサーバーでは機能しますが、他のサーバーでは機能しません

一部のサーバーでは Kerberos 認証が成功しても、他のサーバーでは失敗する場合は、次の領域を調査します。

サーバーが複数の Active Directory ドメインにまたがる場合は、ドメインごとに個別の Kerberos 認証情報を設定します。/etc/krb5.conf ファイルにすべてのレルムのエントリが含まれていることを確認します。各ドメインには、正しいusername@REALMプリンシパルを持つ独自の認証情報が必要です。

動作中のサーバー上の WinRM 設定と障害が発生したサーバーを比較します。各サーバーで次のコマンドを実行します。

```
winrm get winrm/config
```

リモートデスクトップとの接続をテストして、問題を分離します。検出ツールは 形式を使用しusername@DOMAIN、リモートデスクトップは 形式を使用しますDOMAIN\username。

サービスアカウントが、障害が発生したサーバーのローカル管理者グループのメンバーであることを確認します。ターゲットサーバーで次のコマンドを実行します。

```
net localgroup Administrators
```

WMI には、オペレーティングシステム情報にアクセスするためのローカル管理者権限が必要です。SQL Server コレクションでは、サービスアカウントにターゲットサーバーに対するローカル管理者アクセス権も必要です。

Kerberos 設定チェックリスト

データ収集を開始する前に、次のチェックリストを使用して Kerberos 設定を確認します。

- krb5.conf ファイルは検出ツール VM に存在します。
- の領域名は大文字ですkrb5.conf。
- サービスアカウントkinitで を実行すると、エラーなしで成功します。
- を実行すると、有効で有効期限が切れていないチケットklistが表示されます。
- プリンシパル名は Active Directory アカウント名と完全に一致します。
- DNS 解決は KDC ホスト名で機能します。
- ポート 88 の KDC へのネットワーク接続が確認されます。
- ポート 5985 および 5986 のターゲット Windows サーバーへのネットワーク接続が確認されます。
- (マルチドメイン) 各 Active Directory ドメインには、検出ツールで独自の認証情報が設定されており、すべてのドメインの [realms] エントリと [domain_realm] エントリkrb5.confが含まれています。

Oracle Database のトラブルシューティング

データ不足やエラーなど、Oracle Database の収集に関する問題を診断するには、以下を確認してください。

接続が拒否またはタイムアウトしました

症状: Oracle コレクションのステータスは、サーバーの接続エラーを示します。

この問題のトラブルシューティングを行うには、次を確認します。

- Oracle リスナーがターゲットホストで実行されていることを確認します。 `lsnrctl status`
- 検出ツールからポート 1521 (またはカスタムポート) の Oracle ホストへのネットワーク接続を確認します。 `nc -zv <oracle-host> 1521`
- ファイアウォールルールが Oracle リスナーポートでのインバウンド接続を許可していることを確認します。
- Oracle ホスト `lsnrctl services` で を実行して、サービス名を確認します。サービス名が正しくない場合、Oracle リスナーは接続を拒否します。

認証の失敗 (ORA-01017)

症状: 無効なユーザー名またはパスワードエラーで収集が失敗します。

この問題のトラブルシューティングを行うには、次を確認します。

- Oracle サービスアカウントが存在し、ロックされていないことを確認します。 `SELECT account_status FROM dba_users WHERE username = 'DISCOVERY_USER';`
- 手動で接続して、パスワードが正しいことを確認します。 `sqlplus discovery_user/<password>@<host>:1521/<service_name>`
- アカウントがロックされている場合は、ロックを解除します。 `ALTER USER discovery_user ACCOUNT UNLOCK;`

権限が不十分 (ORA-01031)

症状: 接続は成功しますが、コレクションは不完全なデータを返します。

この問題のトラブルシューティングを行うには、次を確認します。

- `SELECT_CATALOG_ROLE` が付与されていることを確認します。 `SELECT * FROM dba_role_privs WHERE grantee = 'DISCOVERY_USER';`
- 不足している場合は、必要なロールを付与します。 `GRANT SELECT_CATALOG_ROLE TO discovery_user;`

手動認証情報はエラーを表示しますが、自動接続は機能します

認証情報を手動でサーバーに固定しても、接続が失敗しても検出ツールはフォールバックしません。認証情報で設定したポートとサービス名が、その特定のサーバーの Oracle リスナーと一致していることを確認します。サーバーに非標準のポートまたはサービス名がある場合は、それに応じて認証情報設定を更新します。

OS レベルのフォールバックが Oracle を検出しない

データベース認証情報が設定されておらず、OS レベルのフォールバックが Oracle を検出しない場合:

- SSH または WinRM OS 認証情報が設定され、サーバーに対して動作していることを確認します (OS メトリクスの収集ステータスを確認)。
- Linux ホストの場合は、`/etc/oratab`が存在するか、Oracle プロセスモニター (pmon) プロセスが実行されていることを確認します。
- Windows ホストの場合は、Oracle レジストリエントリ `HKLM\SOFTWARE\Oracle` が存在するか、`oracle.exe` プロセスが実行されていることを確認します。

SNMP のトラブルシューティング

検出ツール VM にアクセスする

- 検出ツール VM にログインし、vCenter でリモートコンソールを開きます。
 - ユーザー名: `discovery`
 - パスワード: `password`

SNMP ツールのインストール (必要な場合)

- `sudo yum install net-snmp-utils -y`

Linux サーバーへの SNMP 接続をテストする

1. `snmptable -v 2c -c <COMMUNITY_STRING> <REMOTE_SERVER_IP> .1.3.6.1.2.1.6.13.1`
2. 例:

```
#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
```

ネットワーク収集エラー

パスワードを読み取るにはターミナルが必要です

エラー:

```
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
```

ss コマンドは、ユーザーパスワードの入力を求めます。設定された ssh ユーザーは `sudoers` グループに属し、`ss/netstat` コマンドのパスワードレス `sudo` で設定する必要があります。パスワードレス `sudo` を設定するには:

1. 新しい `sudoers` ファイルを作成します。

```
sudo vi -f /etc/sudoers.d/<username>
```

2. 行を追加します。

```
<username> ALL=(ALL) NOPASSWD: /usr/sbin/ss, /usr/bin/netstat
```

3. この変更後、`sudo ss -tnap`と を実行する`sudo netstat -tnap`と、パスワードの入力を求められることなく実行されます。

sudo なしで実行されたネットワークコレクション

検出されたインベントリページに次の警告が表示された場合:

```
Network collection ran without sudo. Process-level connection data may be missing.
```

この警告は、SSH ユーザーアカウントにターゲットサーバーでの `sudo` アクセスがないことを示します。`sudo` を使用しない場合、検出ツールは引き続きネットワーク接続データを収集できますが、どのプロセスが各接続を所有しているかを判断することはできません。完全なプロセスレベルの接続データを収集するには、SSH ユーザーがターゲットサーバーで `sudo` アクセスを持っていることを確認します。

OS メトリクス収集エラー

Linux サーバーのサーバー UUID がない

検出ツールが Linux サーバーのサーバー UUID を収集できない場合 (空または欠落と表示)、それらのサーバー用に設定された SSH 認証情報に `sudo` 権限があることを確認します。ツールは `dmidecode` を使用してサーバー UUID を読み込みます。がインストールされ `dmidecode` ていない場合、ツールは の読み取りにフォールバックするため `/sys/class/dmi/id/product_uuid`、`sudo` アクセスも必要です。`sudo` がない場合、どちらのメソッドも UUID を取得できません。

解決策: 検出ツールに提供された SSH ユーザーアカウントに、ターゲット Linux サーバーに対する `sudo` アクセスがあることを確認します。

検出されたインベントリのアクセスの問題

認証情報の欠落やアクセス拒否などのメッセージがサーバーコレクションステータスに表示される場合:

1. 検出されたサーバーのテーブルでサーバーを選択します。
2. アクセス認証情報の管理 以下を選択できます。
 - a. 認証情報の選択ドロップダウンから代替認証情報を選択します。
 - b. 新しい認証情報を使用して新しい認証情報を指定するを選択します。

3. [保存] します。

変更を保存した後、検出ツールは接続を再試行します。

SSH キー認証のトラブルシューティング

検出ツールからの SSH キー接続のテスト

SSH キー認証が失敗した場合は、検出ツールからターゲットサーバーへの接続を確認します。

1. 検出ツールにログインします (vSphere コンソールまたは SSH 経由で Linux ホストにログインします)。
2. プライベートキーを使用して SSH 接続をテストします。

```
ssh -i /path/to/private_key -o StrictHostKeyChecking=no <username>@<target_ip>
```

3. 接続が成功した場合、問題はキーが検出ツールにアップロードされた方法にあります。キーを再アップロードし、ユーザー名が一致していることを確認します。
4. 接続が失敗した場合は、次の表のエラーメッセージを確認してください。

エラーメッセージ	原因	解像度
Permission denied (publickey)	パブリックキーがターゲットサーバーの <code>authorized_keys</code> ファイルにないか、ユーザー名が間違っています。	正しいユーザーのために、ターゲットサーバーの <code>~/.ssh/authorized_keys</code> にパブリックキーを追加します。ファイルアクセス許可の確認: <code>chmod 700 ~/.ssh && chmod 600 ~/.ssh/authorized_keys</code> 。
Connection timed out after 20s	ポート 22 は検出ツールから到達できません。	検出ツールとターゲットサーバーの間でポート 22 が開いていることを確認します。ファイアウォールとセキュリティグループを確認します。
Connection refused	SSH サービスはターゲットサーバーで実行されていません。	SSH サービスを起動します: <code>sudo systemctl start sshd</code> 。

エラーメッセージ	原因	解像度
Invalid SSH key for credential ' <i>name</i> '	キー形式がサポートされていないか、キーデータが破損しているか、パスフレーズがないか、正しくありません。	キー形式が PEM、OpenSSH、または PKCS#8 形式の RSA、ECDSA、または Ed25519 であることを確認します。キーが暗号化されている場合は、パスフレーズが正しいことを確認してください。

Linux インストーラのトラブルシューティング

ポート 5000 は既に使用されています

症状: 検出ツールサービスはインストール後に起動しません。

解決策: ポート 5000 を使用してプロセスを特定して停止します。

```
sudo ss -tlnp | grep :5000
```

競合するプロセスを停止し、検出ツールを再起動します。

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

一般的なエラーメッセージ

この表は、一般的なエラーメッセージとその説明を示しています。

メッセージ	ロケーション	説明
パスワードはすでに作成されています	パスワードの作成ページ	2 人のユーザーが同時にパスワードを作成する場合のレース条件。更新
エクスポートに失敗しました	インベントリページ	ログの再試行または送信
オンデマンドコレクションはすでに進行中です	インベントリページ	2 人のユーザーが同時に手動収集を開始したときのレース条件。現在の手動収

メッセージ	ロケーション	説明
		集が終了したら、もう一度試してください。
Command timed out after 60s	サーバーコレクションのステータス	ターゲットサーバーのコマンドが 60 秒以内に完了しませんでした。これは、負荷の高いサーバーで発生する可能性があります。コレクションを再試行するか、ターゲットサーバーの負荷を調査します。
1 つ以上の認証情報に不明な UUIDsが含まれている	OS アクセスページ	2 人のユーザーが同時に OS 認証情報を編集する場合のレース条件。もう一度試してください
無効なパスワード	サインインページ	ログイン用のパスワードが正しくありません。管理者に問い合わせるか、お問い合わせください
セッションの有効期限が切れています。再度ログインしてください。	サインインページ	セッションがタイムアウトしました。再度ログインする必要があります
内部エラーが発生しました	さまざまなページ	ログの再試行または送信

リリースノート

このセクションでは、AWS Transform 検出ツールの機能、改善点、バグ修正をリリース日別に整理して説明します。

日付	バージョン	リリースノート
2026 年 6 月 30 日	1.0.1646	新しい特徴 Oracle Database 検出を追加しました。このツールは、CDB および PDB トポロジ、バージョン、エディション、イン

日付	バージョン	リリースノート
		ストールオプション、機能の使用、およびコンポーネントインベントリを収集します。検出ツールは、直接 SQL 接続 (SELECT_CATALOG_ROLE を使用) と SSH または WinRM による OS レベルのフォールバック検出をサポートします。 - 異なる Active Directory ドメイン間で複数の Kerberos 認証情報のサポートが追加されました。各認証情報は個別に認証され、認証の競合なしで複数のドメインの Windows サーバーから収集できます。 改良点 - レガシーホストの OS メトリクス収集が改善されました。最新のコマンドが使用できない古いオペレーティングシステム (RHEL 5、SLES 11、Windows Server 2008 R2) のフォールバックデータ収集を追加しました。ストレージ、パフォーマンス、ネットワークインターフェイスのデータが、代替システムインターフェイスを使用して収集されるようになりました。 バグ修正 - Hyper-V を介して検出されたサーバーの Kerberos 認証の失敗を修正しました。検出ツールは、Kerberos SPN マッチングに必要な Hyper-V ゲストメタデータからサーバーホスト名 (FQDN) を正しく解決するようになりました。

日付	バージョン	リリースノート
2026 年 5 月 14 日	1.0.1363	新しい特徴 • サポートされている Linux ディストリビューション (Amazon Linux 2、Amazon Linux 2023、RHEL 8～9、Rocky Linux 8～9、AlmaLinux 9、Ubuntu 20.04～24.04、Debian 10～12、SLES 15 SP5) に検出ツールをデプロイするための Linux インストーラを追加しました。 • vCenter 間で自動 VM 重複排除を使用して複数の VMware vCenters。 • Linux OS アクセスのユーザー名とパスワードの代わりに SSH プライベートキー認証を追加しました。サポートされているキー形式には、RSA、ECDSA、Ed25519、OpenSSH、PKCS#8 などがあります。 • 検出されたインベントリにない RFC 1918 プライベート IP アドレスとの間の接続をキャプチャするプライベートアドレスネットワークコレクションを追加しました。 • カスタム日付範囲を選択し、エクスポートに含めるデータモジュールを選択するエクスポートオプションを追加しました。 改良点 • コレクタースケジュールをずらすことで、スケジュールされたコレクションの信頼性が向上しました。 • SSH コマンドバッチ処理により OS メトリクスの収集パフォーマンスが向上しました。 • 認証情報の変更は、以前に障害が発生したサーバーをすぐに再試行するようになりました。 • 一部のコマンドが空の結果を返すと、OS メトリクス収集が部分的なステータスを報告するようになりました。 • ダッシュボードでの vCenter 接続の問題のエラーの可視性が向上しました。

日付	バージョン	リリースノート
2026 年 4 月 7 日	1.0.1074	新しい特徴 - フェイルオーバークラスター重複排除を使用した Hyper-V ホストからの VMs の自動検出など、Microsoft Hyper-V 検出のサポートが追加されました。 - CSV ファイルのアップロードによるサーバーインポートのサポートが追加されました。 - サーバーパフォーマンス、ストレージパフォーマンス、サーバープロビジョニング、ストレージプロビジョニング、ネットワークインターフェイス、実行中のプロセスという 6 つの独立したモジュールを含む OS メトリクスコレクションを追加しました。 - VHD イメージファイルを使用した Hyper-V デプロイオプションを追加しました。 - EC2-specificサービスを削除し、インスタンスメタデータエンドポイントへのアクセスをブロックすることで、オンプレミスのセキュリティ強化が改善されました。
2026 年 3 月 26 日	1.0.1014	バグの修正と改善 - Kerberos ドメイン大文字と小文字の検証をバックエンドからフロントエンドに移動しました。 - Python-kerberos から Python-pykerberos に切り替え、ファイルベースのキャッシュをデフォルトにしました。 - 大文字と小文字を区別しないサーバー ID の一括編集の検証を修正しました。 - エクスポート ZIP ファイル名 (現在は) を修正しましたdiscovery_tool_export.zip 。 - チェックを実行するサービスを修正し、システム認証情報のデバッグログを追加しました。 - アーティファクトを構築するためのネットワーク設定スクリプトを追加しました。

日付	バージョン	リリースノート
2025 年 12 月 16 日	1.0.718	新しい機能と改善点 • ダウンストリームサービスでの VM の重複を防ぐための SMBIOS UUID サポートを追加しました。 • Min/Max/Avg 追跡を導入し、リソースモニタリングを改善するためのストレージメトリクスを拡張しました。 • API 検証の改善を追加しました。 • コンテキストガイダンスを含むヘルプパネル機能を追加しました。 • インターフェイス全体で用語を「コレクター」から「ツール」に更新しました。
2025 年 10 月 31 日	1.0.0	初回リリース AWS Transform 検出ツールの初回起動。このリリースには以下が含まれます。 • オフライン収集 (AWS アカウントまたは Application Discovery Service への依存なし) 。 • VMware の検出とモニタリング。 • SQL Server データベースの検出。 • SSH、WMI、SNMP による Server-to-server 通信の収集。

AWS 変換コネクタ

AWS 変換コネクタを使用すると、移行とモダナイゼーションのワークフローのために、アカウントの境界を越えてリソースに安全にアクセスできます。これにより、セキュリティコントロールとアクセス許可を維持しながら、アカウントの境界を越えてリソースにアクセスして管理できます。コネクタは、AWS 変換が有効なソースアカウントと、AWS ターゲットアカウントまたはサードパーティシステムの外部リソース間の接続を表します。

コネクタには、ソースコネクタからのアクセスリクエストと、ターゲットアカウントの所有者からの承認が必要です。このリクエストは、ソースコネクタを設定するときに行います。同様に、他のアカウントが自分のアカウントにアクセスできるように、送信先コネクタを承認するリクエストを受け取る場合があります。

- コネクタの作成: AWS Transform が有効なソースアカウントにコネクタを作成し、ターゲットアカウントと必要なアクセス許可を指定します。
- アクセス許可の設定: AWS Transform では、移行先アカウントで移行アクションを実行するために、次のような特定の IAM ロールとアクセス許可が必要です。
 - を使用した移行データの管理 AWS KMS
 - クロスリージョン API コールの実行
 - VMware サーバーへのレプリケーションエージェントのインストール
 - ネットワークインフラストラクチャ (VPCs、サブネット、ルーティング) の作成
 - Amazon EC2 インスタンスの起動と CloudFormation スタックのデプロイ
 - Migration Hub を使用した移行ワークフローの実行
- 承認プロセス: Transform がリソースにアクセスする前に AWS 、送信先コネクタリクエストを承認する必要があります。
- アクティブ接続: コネクタを承認すると、AWS 変換ジョブがターゲットアカウントのリソースに安全にアクセスして管理できるようになります。

エージェントオペレーションのユーザートレサビリティ

デフォルトでは、は AWS 、変換サービスアイデンティティで AWS リソースに対するエージェントアクション AWS CloudTrail を記録します。ユーザートレサビリティは、ジョブを開始または操作した特定の AWS IAM Identity Center ユーザーにこれらのアクションを属性付けし、個々のユーザーレベルでセキュリティ監査とインシデント調査を可能にします。

Note

ユーザートレサビリティは、IAM Identity Center (IDC) ユーザーでのみサポートされています。

ユーザートレサビリティを有効にするには、AWS 変換プロファイル設定でバックグラウンドセッションを有効にします。有効にすると、AWS Transform はジョブを操作するときに IAM アイデンティティセンターの[バックグラウンドセッション](#)を自動的に作成し、それを使用してコネクタオペレーション用のアイデンティティ拡張 IAM ロールセッションを生成します。これらのセッションでは、ダウンストリーム AWS サービス呼び出しと AWS CloudTrail イベントにユーザー ID が埋め込まれます。バックグラウンドセッションが無効になっている場合、AWS Transform はユーザー ID コンテキストなしで標準引き受けロールセッションにフォールバックします。

新しいコネクタは、ユーザーのトレサビリティをサポートするように自動的に設定されます。既存のコネクタの場合、コネクタロールの信頼ポリシー `sts:SetContext` に を手動で追加する必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": ["transform.amazonaws.com"] },
      "Action": ["sts:AssumeRole", "sts:SetContext"],
      "Condition": {
        "StringEquals": { "aws:SourceAccount": "your-account-id" }
      }
    }
  ]
}
```

ジョブページのバックグラウンドセッションタブから、アクティブなバックグラウンドセッションを表示および取り消すことができます。セッションを取り消すと、セッションとそのセッションから派生したコネクタセッションが直ちに無効になります。

詳細については、「AWS IAM アイデンティティセンターユーザーガイド」の「[Trusted identity propagation overview](#)」を参照してください。

コネクタの管理

コネクタページには、以下が表示されます。

- Source Connectors タブ: 他のアカウントに接続するために作成したすべてのコネクタを一覧表示し、ターゲットアカウントとコネクタのステータスを表示します。
- Destination Connectors タブ: リソースへのアクセスを希望し、承認または拒否を要求する他のアカウントからの受信コネクタリクエストを表示します。

AWS 変換では、アクティブな移行ワークフローを中断する可能性のある偶発的な変更を避けるために、コネクタの承認、拒否、削除などの重要なアクションを確認する必要があります。

コネクタを作成するには

1. Source Connectors タブに移動します。
2. ターゲットアカウントとリソースタイプを指定します。
3. 必要なアクセス許可とアクセス範囲を設定します。

アクセス許可を設定するには

1. 必要なアクセス許可を持つ新しい IAM ロールを作成するか、
2. コネクタ用に以前に作成した既存のロールを選択します。
3. ロールに、AWS 変換オペレーションに必要なすべてのアクセス許可があることを確認します。

受信リクエストを管理するには

1. Destination Connectors タブでコネクタリクエストを確認します。
2. 正当なリクエストを承認して、クロスアカウントアクセスを有効にします。
3. 許可されていない、または不要な接続の試行を拒否します。

コネクタを管理するには

1. アクティブなコネクタのセキュリティとコンプライアンスをモニタリングします。
2. 不要になったコネクタを削除します。これにより、コネクタを使用したジョブの実行が失敗することに注意してください。

- 要件の変化に応じてアクセス許可を更新します。

移行 (VMware を含む)

AWS 変換は、VMware、Hyper-V、その他のソースなどの生成 AI を使用して、仮想サーバー環境とベアメタルサーバー環境を Amazon EC2 に移行するのに役立ちます。この章では、移行機能の AWS 変換と移行プロセスのワークフローの概要を説明します。

機能と主要機能

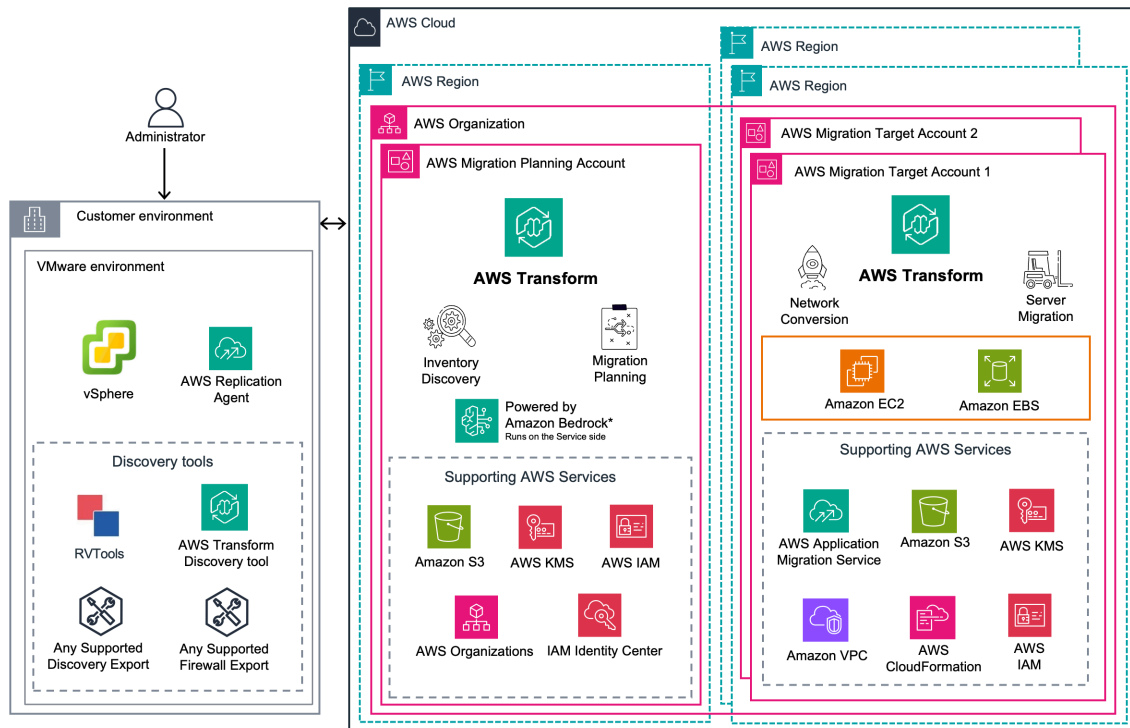
AWS Transform には、サーバー環境を に移行するための以下の機能と主要な機能があります AWS。

- 複数の検出オプション:
 - からのコレクターを使用して、VMware 環境の検出を支援しました AWS Application Discovery Service。
 - オープンソースの [Export for vCenter](#) ツール (VMware 環境) を使用します。
 - 個別に収集された検出データ (任意のソース環境) をインポートします。
- ソースネットワーク設定を Amazon VPC ネットワークアーキテクチャに AI 主導で変換します。
- アプリケーションのグループ化や推奨される移行ウェブなど、AI 主導の移行計画の生成。
- Amazon EC2 でネイティブに実行するようにサーバーをリホストします。
- ローカライズされたウェブアプリケーションインターフェイス。任意の言語で移行に AWS Transform を使用できます。詳細については、「[言語設定](#)」を参照してください。

AWS Transform は、サポートされているオペレーティングシステムの Windows および Linux サーバーの移行をサポートしています。サポートされているオペレーティングシステムの完全なリストについては、「AWS Transform MGN ユーザーガイド」の「[サポートされているオペレーティングシステム](#)」を参照してください。

AWS 移行アーキテクチャの変換

この図は、移行アーキテクチャの AWS 変換の概要を示しています。



移行ジョブ

移行に AWS Transform を使用するには、まずワークスペースが必要です。ワークスペースは、1 つ以上の変換ジョブを作成できる論理コンテナです。このトピックのセクションでは、ワークスペースを取得する方法と、ワークスペースで移行ジョブを作成して開始する方法について説明します。

ワークスペースの取得

ワークスペースの取得の詳細については、[「開始方法」](#)を参照してください。

使用するワークスペースによって、変換ジョブを作成できる AWS リージョンが決まります。これは、ジョブが存在する AWS リージョンです。検出データと AWS 変換のレコメンデーションもこれに含まれます AWS リージョン。別の にワークスペースとジョブを作成するには AWS リージョン、管理者に別のワークスペースを作成するように依頼してください。サポートされる の詳細については AWS リージョン、「」を参照してください [サポート対象のリージョン](#)。

ジョブを作成し、検出データとレコメンデーションを保存 AWS リージョン できる は AWS Transform 管理者によって決定されますが、移行のターゲット AWS リージョンとして別の を指定できます。つまり、検出を実行して 1 つの で AWS Transform レコメンデーションを受け取り AWS

リージョン、別のターゲット環境を作成できます AWS リージョン。その場合は、データを転送します AWS リージョン。詳細については、「[the section called “ターゲットの AWS アカウントとリージョンを接続する”](#)」を参照してください。

ジョブタイプ

AWS Transform には、移行のニーズに応じて選択できる次のタイプの移行ジョブが用意されています。これらのプリセットオプションに加えて、ジョブの任意のステップをいつでも動的に追加または削除して、移行ワークフローをカスタマイズできます。

End-to-end移行

1. 検出を実行する
2. 移行計画を立てる
3. ターゲットアカウントを接続する
4. ランディングゾーンを構築する
5. ネットワークの移行
6. サーバーの移行

検出と移行の計画

1. 検出を実行する
2. 移行計画を立てる

ネットワーク移行

1. ターゲットアカウントを接続する
2. ネットワークの移行

ランディングゾーン

1. ターゲットアカウントを接続する
2. ランディングゾーンを構築する

ランディングゾーン、ネットワーク、サーバーの移行

1. ターゲットアカウントに接続する
2. ランディングゾーンを構築する
3. ネットワークの移行
4. サーバーの移行

移行計画とサーバー移行

検出、ウェーブプラン、リホストが含まれます。

1. 検出を実行する
2. 移行計画を立てる
3. ターゲットアカウントに接続する
4. サーバーの移行

ジョブの作成と開始

移行プロジェクトの最初のステップは、AWS 変換ジョブを作成することです。移行プロジェクトでは、目標に応じてさまざまなジョブタイプを選択できます。次の手順では、任意のタイプの新しい移行ジョブを作成して開始する方法について説明します。さまざまなジョブタイプの詳細については、「」を参照してください[the section called “ジョブタイプ”](#)。

新しい移行ジョブを作成して開始するには

1. ワークスペースのランディングページで、ジョブの作成を選択します。
2. 移行オプションを選択し、作成する移行ジョブのタイプを指定します。各移行ジョブタイプに含まれる手順については、「」を参照してください[the section called “ジョブタイプ”](#)。
3. チャットのすべての質問に回答したら、ジョブの作成を選択します。

ワークスペースの概要レポートのダウンロード

ワークスペースの概要レポートは、移行中いつでもダウンロード可能な PDF として生成できます。このレポートには、ジョブのステータスと現在のワークフローステップ、ユーザーのアクションと承認、ウェーブプランニングの詳細、ネットワーク移行トポロジ、ランディングゾーン設定、リホスト

の進行状況、コンテナ化の決定、生成された主要なアーティファクトなど、ワークスペース内のすべての移行ジョブが統合されたビューが表示されます。レポートを生成するには、チャットのエージェントにワークスペースの概要を尋ねます。たとえば、「すべてのジョブにおける移行の進行状況の概要をワークスペースに渡す」と入力します。エージェントはすべてのジョブからデータをコンパイルし、PDF をダウンロード可能なファイルとしてチャットに直接配信します。このレポートを使用して、全体的な移行の進行状況を追跡したり、ステークホルダーとステータスを共有したりできます。

制限事項

AWS 変換には次の制限があります。

- 実行中の移行ジョブを停止し、エージェントに再起動を依頼すると、ジョブは最初から再開され、ジョブで行った進行状況はすべて失われます。ただし、再起動前にジョブで作成されたアーティファクトは引き続き使用できます。
- 移行ジョブ AWS リージョン ごとに 1 つのターゲットを指定できます。アプリケーションを異なるターゲットリージョンに移行するには、複数の移行ジョブを作成します。
- マルチアカウント移行 – 単一リージョンのみ – 単一内で複数のアカウントに移行できます AWS リージョン。マルチリージョン移行の場合、ターゲットリージョンごとに個別のプロジェクトを作成する必要があります。
- マルチアカウント移行 – ウェーブごとに 1 つのアカウント – 各移行ウェーブは 1 つのアカウントのみをターゲットにできます。異なるターゲットアカウントを必要とするアプリケーションは、別々のウェーブに配置する必要があります。
- マルチアカウント移行 – 必要な AWS 組織 – すべてのターゲットアカウントは AWS 組織の一部である必要があります。

ソースデータを検出する

オンプレミスデータを検出して収集するには、[AWS Transform discovery ツール](#)、AWS Migration Evaluator collector、または RVTools と ModelizeIT からのエクスポートのアップロード、および Cloudamize などのツールによって生成された AWS Migration Portfolio Assessment (MPA) 形式のエクスポートを使用できます。AWS Transform は、AWS Transform discovery ツールのエクスポートを ZIP ファイル形式で CSV および JSON ファイルとして処理し、RVTools は CSV ファイルを含む Excel ファイルまたは ZIP ファイルとしてエクスポートし、ModelizeIT CSV エクスポートを ZIP ファイルで処理できます。データソースごとに、AWS Transform はプライマリサーバーファイルを個別に受け入れます。ただし、接続などの補足ファイルは、サーバーファイルを含む ZIP ファイルまたはサーバーシートを含む Excel ファイルに含まれている場合にのみ処理されます。

データの完全性と正確性を確保するために、利用可能な最も詳細なデータをアップロードし、アップロード前にエクスポートファイルを確認することをお勧めします。必要なすべてのファイルがアップロードに含まれていることを確認し、データが現在の環境状態を反映していることを確認します。この予備レビューは、再アップロードの必要性を最小限に抑え、AWS Transform がオンプレミス環境をより正確にキャプチャし、アプリケーションのグループ化やウェーブプランニングを含む移行計画をより適切にサポートできるようにします。より良いデータを取得するにつれて、データを段階的に追加することもできます。

AWS 変換は、ファイル形式と構造を自動的に検出し、構造化されたエンティティレコードを解析して抽出し、複数のファイル間で重複を削除し、データ品質を検証して問題を報告し、ダウンストリーム移行計画の準備を整えます。

オンプレミスデータを共有して確認したら、ジョブプランでオンプレミスデータの検出を展開し、インベントリの準備状況の概要を選択することで、検出データを確認できます。アップロードしたデータについて質問して、AWS 変換がすべてを正しく取り込んだことを確認するか、データ処理の誤りを特定して修正することもできます。たとえば、オペレーティングシステムとバージョンについて質問できます。誤りを修正する必要がある場合は、使用した検出ツールからインベントリデータを再アップロードできます。AWS Transform は更新されたレコードを自動的に処理、複製解除、マージします。そのインベントリデータを使用しなくなった場合は、以前にアップロードしたファイルを削除することもできます。

移行計画を立てる

AWS Transform for VMware 内の移行計画ジョブステップは、大規模な移行を計画するための共同チャットベースのエクスペリエンスです。AWS Transform エージェントは AWS、規範ガイダンスを適用して、オンプレミスデータの分析から最終的な移行ウェーブプランまで、顧客をガイドします。

オンプレミスデータ検出ジョブが正常に完了すると、AWS Transform は検出データを使用してアプリケーションを移行ウェーブにグループ化します。AWS Transform は、サーバーの分析とスコープ、アプリケーションへのグループ化、移動グループの生成、移行ウェーブの構築のステップをガイドします。オンプレミス環境を分析するときは、AWS Transform がインストール済みソフトウェアをどのように分析したかをよりよく理解するために質問できます。たとえば、サーバーの依存関係やネットワークアーキテクチャなどです。

AWS 変換は、アプリケーショングループとウェーブ内のスコープ調整をサポートします。検出データはいつでも再アップロードでき、AWS Transform は新しいレコードを自動的に処理、複製解除し、既存のデータとマージします。新しく検出された依存関係やインフラストラクチャの追加など、

変更が検出されると、AWS Transform は影響を受ける依存関係グループにフラグを付け、ウェーブプランの調整に関する推奨事項を提供します。移行計画では、非構造化テキストデータを活用して計画プロセスを強化することもできます。

移行計画には 4 つの段階があります。

- スコープと分析では、検出データを確認し、ソフトウェアとネットワーク環境について質問し、移行の対象となるリソースを決定できます。
- グループアプリでは、ホスト名分析、ネットワークの依存関係、ビジネスルールなどに関するビジネスルールと技術ルールを組み合わせて提供することで、移行計画でインフラストラクチャをアプリケーションにグループ化できます。アプリケーションのインベントリが既にある場合、移行計画では代わりにそのインベントリを使用できます。
- 移動グループの生成では、移行計画に技術要件とビジネス要件を提供して、一緒に移動する必要があるアプリケーションを決定できます。技術的な依存関係には、データベース、メッセージキュー、または複数のアプリケーション間で共有されるその他のリソースが含まれます。ビジネスと運用の依存関係には、ビジネスの重要性、RPO と RTO、データセンターの場所、アプリケーション所有者が含まれます。
- 最後に、ビルドウェーブでは、タイムラインと優先順位に関する移行計画のコンテキストを提供して、移行できるウェーブ計画を構築できます。優先順位スコア、移動グループサイズ、ユーザー数、アプリケーションの複雑さなどの要因に基づいて、ウェーブに含める移動グループを選択できます。

Migration Planning の用語:

- 移行ウェーブは、一緒に移行される論理グループです。移行ウェーブは、1 つ以上の移動グループで構成されます。
- 移動グループは、一緒に移動する必要がある共依存アプリケーションのセットです。共有データベースなどの技術的な依存関係がある場合や、共有ビジネス機能のサポートなどのビジネス依存関係がある場合があります。
- 依存関係はシステム間の関係です。依存関係には、次のようないくつかのタイプがあります。
 - システムが依存関係なしでは動作しない、クリティカルまたはハードな依存関係。この一般的な例は、データベース、他のアプリケーション、またはサービスに依存するアプリケーションです。
 - ソフト依存関係。システムのオペレーションには重要ではありません。一般的な例としては、個別に移行できるレイテンシーの影響を受けない依存関係などがあります。

- 非技術的な依存関係には、ビジネス、組織、運用、コンプライアンスの依存関係が含まれます。これらは、組織とその優先順位に関連する依存関係です。例としては、ビジネス機能の共有や組織の所有権などがあります。

ワークフロー

移行計画はインタラクティブで反復的なワークフローです。戻ると、いつでも前のステップを変更できます。一般的な移行計画ワークフローは次のとおりです。

1. 移行計画は、利用可能な検出データを要約することから始まります。利用可能なデータを確認し、いつでも検出ステップに戻って追加のデータを提供します。
2. スcopeと分析のステップでは、オンプレミス環境について質問して、収集したデータを検証できます。質問の例は次のとおりです。
 - a. オペレーティングシステム別にサーバーを一覧表示する
 - b. オンプレミスのネットワークポートロジを要約する
 - c. 環境で実行されている最も一般的なテクノロジーを一覧表示します。
3. 環境の分析中に、移行対象外のサーバーを特定した場合、それらのリソースを除外するように AWS Transform に指示できます。これには、次のような例があります。
 - a. ホスト名にレガシーを持つすべてのサーバーを削除する
 - b. 10.0.2.0/24 サブネット内のすべてのサーバーを削除する
 - c. 2022 より前のバージョンの Windows を実行しているすべてのサーバーを削除する
4. 環境を十分に探索し、移行範囲を決定したら、AWS Transform に次の移行計画ステップに移行するように指示できます。
5. 次のステップは、アプリケーションのグループ化です。サーバーが既にアプリケーションにマッピングされている場合は、そのマッピングを使用するように AWS Transform に指示し、このステップをスキップできます。アプリケーションが事前に定義されていない場合は、アプリケーションを定義する技術ロジックとビジネスロジックを提供できます。AWS Transform はアプリケーションのグループ化プロセスをガイドし、サーバーをアプリケーションに効果的にグループ化するために提供できるデータポイントを提案します。オンプレミスアプリケーションに関して提供できる情報が多いほど、AWS Transform はより効果的にサーバーをアプリケーションにグループ化できます。十分な情報を提供したら、AWS 変換にアプリケーションのグループ化を実行するように指示できます。
6. アプリケーションのグループ化が実行されたら、アプリケーショングループを確認します。変換には、次のような必要な変更を加える AWS 変換のように指示できます。

- a. サーバーの example-server を application-5 に移動する
 - b. application-5 "HR App Test Environment" の名前を変更する
 - c. IIS Dev Farm からすべての Linux サーバーを削除する
7. アプリがグループ化されたら、AWS 変換に次のステップに進むように指示します。
8. 次のステップは移動グループ化です。移動グループ化ステップでは、一緒に移動する必要があるアプリケーションを特定します。技術的依存関係と非技術的依存関係に関するコンテキストを提供します。AWS Transform はプロセスをガイドし、アプリをグループ化するために提供できるデータポイントを提案します。この段階では、いくつかの考慮事項があります。
- a. 移動グループのターゲットサイズはどのくらいですか？
 - b. 開発、テスト、製品などの環境をアプリごとに組み合わせるか、分割するか。
 - c. ネットワークの依存関係をどのように考慮しますか？すべての依存関係は重要ですか、または一部の依存関係はソフト依存関係と見なされ、移動グループ間で分割できますか？
9. 移動グループ化のルールを指定したら、移動グループ化戦略を実行するように AWS Transform に指示します。その後、移動グループを確認して変更できます。移動グループを確認したら、最終的な移行計画ステップに移行するように AWS Transform に指示できます。
10. ウェーブプランニングは、移行計画の最後のステップです。このステップでは、移動グループを移行ウェーブにグループ化し、それらのウェーブに優先順位を付けます。ウェーブプランニングステップ内で、AWS Transform は、移動グループをウェーブにグループ化し、それらのウェーブに優先順位を付けるために必要なビジネスの優先順位付けを提供する方法を説明します。ウェーブプランニングにおける考慮事項は次のとおりです。
- a. 各移動グループのビジネス重要度
 - b. 各移動グループの移行タイムラインとタイムライン
 - c. 各移動グループに関連するリスク
 - d. ウェーブごとに移行するサーバーの数
11. ウェーブにグループ化する方法に関する十分なガイダンスを提供したら、ウェーブ計画を実行するように AWS Transform に指示します。その後、ウェーブを確認して変更することができます。
12. ウェーブプランを確定したら、移行計画を完了して実行に移すことができます。いつでも移行計画に戻って、計画を絞り込み、反復することができます。
13. ウェーブごとに、移行戦略を割り当てることができます。リホスト (サーバーを Amazon EC2 に移行) とコンテナ化 (ソースコードをコンテナ化し、Amazon Elastic Container Service または Amazon Elastic Kubernetes Service にデプロイ) です。戦略コンテナ化するウェーブを割り当てると、AWS Transform は移行の実行中にそのウェーブのソースコードコンテナ化ワークフローを実行します。詳細については、「[ソースコードのコンテナ化](#)」を参照してください。7Rs フレーム

ワーク全体で AWS 推奨戦略を割り当てる前に取得するには、「」を参照してください [the section called “7Rs レコメンデーション”](#)。7Rs

移行戦略 (7R) の推奨事項

移行計画で、AWS Transform は検出されたインベントリを分析し、すべてのサーバーとアプリケーションに対して業界標準の 7Rs つである移行戦略を推奨できます。各レコメンデーションには、推奨される AWS ターゲットサービス、信頼スコア、およびその背後にある推論が含まれます。これにより、ウェーブプランの防御可能な開始点が得られ、見逃す可能性のあるモダナイゼーションの機会が表示されます。

7 つの戦略は、リホスト、リプラットフォーム、リファクタリング、再購入、廃止、保持、再配置です。AWS Transform は、オペレーティングシステム、使用率、ワークロードタイプなどのデータ内のシグナルを読み取り、説明可能なレコメンデーションを作成します。たとえば、アイドルサーバーまたは end-of-life サーバーの使用停止、SQL Server またはオープンソースデータベースの Amazon RDS への移行、E メールおよびコラボレーションワークロードの Microsoft 365 への移行、特殊なレガシーシステムの保持、Amazon EC2 での残りのワークロードのリホストを推奨できます。レコメンデーションは、個々のサーバーからアプリケーション、ポートフォリオ全体にロールアップされ、戦略とウェーブレベルの概要別に重要度が表示されます。

移行戦略 (7R) レポートを生成するには:

1. オンプレミスデータの検出ステップを完了し、サーバーをアプリケーションにグループ化します。AWS Transform がサーバーごとおよびアプリケーションごとのレコメンデーションを提供できるようにするには、両方が必要です。
2. 移行計画チャットで、AWS Transform に移行戦略 (7R) レポートを依頼します。たとえば、R 戦略レポートを生成します。アプリケーションのグループ化が完了すると、AWS Transform はレポートの生成も提供します。
3. AWS Transform はインベントリを分析し、7Rs フレームワークを適用して、レポートを生成します。組織名やフィルターなどの詳細を指定して、分析の範囲を絞り込むことができます。
4. 推奨事項を確認します。レコメンデーションを調整するには、チャットで AWS Transform に指示します。1 つのサーバーまたはアプリケーションを変更するか、一括で変更を適用します (オペレーティングシステム、環境、ウェーブ、優先度など)。AWS Transform は、変更を加えてレポートを更新します。

デフォルトでは、AWS Transform はフィルタリングして探索できるインタラクティブな HTML ダッシュボードを生成します。リクエストに応じて、利害関係者と共有するための PDF ドキュメント、

または提示する Microsoft PowerPoint (PPTX) スライドデッキを生成することもできます。PDF 出力と PowerPoint 出力は、同じ分析の point-in-time スナップショットです。このレポートには、すべてのサーバーとアプリケーションに推奨される戦略と AWS ターゲットサービス、信頼スコアと推論、強調表示されたモダナイゼーションの機会、戦略別の重要度ビュー、ウェーブレベルの概要、リスクフラグが含まれます。

このレポートで推奨される戦略は、ウェーブプランニングで割り当てるウェーブごとの戦略を補完します。ソースコードのコンテナ化については、「」を参照してください [ソースコードのコンテナ化](#)。

ダイアグラムとレポート

移行計画中、AWS Transform は検出されたインベントリと計画結果をビジュアル図と分析レポートに変換します。計画チャットで会話で生成します。たとえば、「ネットワークトポロジ図を表示する」や「リスクレポートを生成する」などです。AWS Transform は、既に指定したデータからビルドします。セットアップは必要ありません。

図はインタラクティブで、環境を視覚的に調べることができます。使用可能な図は次のとおりです。

- ネットワークトポロジ – 数千のノードの大規模な環境にスケールするように設計された、サーバーとそのネットワーク関係のインタラクティブマップ。
- アプリケーションの依存関係 – 観測されたネットワーク接続から導き出された application-to-application 関係。
- Wave Gantt チャート – スケジュール、依存関係、マイルストーン、進行状況を示す移行ウェーブのタイムラインビュー (ウェーブプランニング後に利用可能)。
- 一般グラフ – 選択したデータのディメンションに関する円グラフ、棒グラフ、ヒストグラム、ツリーマップ、サマリーテーブル。

レポートは、移行データに関する説明分析と推奨事項を提供します。使用可能なレポートは次のとおりです。

- リスク評価 – アプリケーションごと、ウェーブごと、ポートフォリオ全体の概要を使用して、技術的な複雑さ、依存リスク、移行準備状況、運用リスクにわたって各アプリケーションをスコアリングします。
- 移行戦略 (7R) – サーバーごとおよびアプリケーションごとの戦略の推奨事項。詳細については、「[the section called “7Rs レコメンデーション”](#)」を参照してください。
- 一般レポートとカスタムレポート – 実行準備の概要、cost-versus-risk のトレードオフ、コンテナ化候補の評価などのリクエストの分析レポート。

一部の出力は、計画している範囲によって異なります。たとえば、アプリケーションの依存関係図では、ネットワーク接続データを使用します。ウェーブガントグラフとウェーブレベルレポートの概要は、ウェーブ計画の完了後に利用可能になります。前提条件がない場合、AWS Transform は必要なものを指示し、最初に必要なステップを実行できます。

AWS Transform は、フィルタリングして探索できるインタラクティブな HTML として図を提供します。デフォルトでは、AWS Transform はレポートを自己完結型の HTML ファイルとして配信します。リクエストに応じて、AWS Transform は利害関係者と共有するための PDF ドキュメント、または提示する Microsoft PowerPoint (PPTX) スライドデッキを生成することもできます。PDF 出力と PowerPoint 出力は、同じ分析の point-in-time スナップショットです。

ターゲットの AWS アカウントとリージョンを接続する

ネットワーク移行、ランディングゾーン構築、サーバー移行用にターゲット AWS アカウント コネクタを設定します。これには、移行タイプの選択、MAP 契約の詳細の提供 (該当する場合)、コネクタのセットアップの 3 つのステップが含まれます。これらの設定は、ネットワーク移行、ランディングゾーン、サーバーリホストなど、すべての移行ステージに適用されます。

ステップ 1: 移行タイプの選択

シングルアカウント移行とマルチアカウント移行のどちらを実行するかを選択します。

- シングルアカウント移行 – すべてのワークロードが 1 つのターゲットに移行します AWS アカウント。コネクタターゲットアカウントとターゲットアカウントは同じです。
- マルチアカウント移行 – ワークロードは異なるターゲットアカウントに移行します。コネクタは、組織管理アカウント、またはと CloudFormation StackSets の両方 AWS Transform MGN に登録されている委任管理者 (DA) アカウントに接続する必要があります。

ステップ 2: MAP 契約

移行が AWS Migration Acceleration Program (MAP 2.0) の一部である場合は、移行ポートフォリオエクスペリエンス (MPE) ID を指定します。これは、大文字と数字 (ABCDE12345 など) を使用した 10 文字のコードです。MPE ID を指定すると、MAP タグがネットワーク移行、ランディングゾーン、サーバーリホストの各ステージで作成されたすべてのリソースに適用されます。タグ形式は次のとおりです。

- キー: map-migrated 値: migMPE_ID

MAP クレジットを受け取るには、MAP タグを適用する必要があります。MAP の詳細については、[AWS「移行促進プログラム」](#)を参照してください。

ステップ 3: コネクタ設定

ターゲットアカウントコネクタを使用して、移行ジョブを移行後にワークロードが存在する AWS 環境に接続します。開始する前に、移行されたインフラストラクチャをサポートするために必要なアクセス許可、クォータ、設定 AWS アカウント がターゲットにあることを確認します。

コネクタリクエストを承認すると、AWS 変換アクセス許可が付与されます。

- VMware 移行用の Amazon S3 バケットオペレーション (読み取り/書き込み) と、AWS Migration Hub および AWS Application Migration Service (MGN) へのアクセスを管理します。これには、次の項目のアクセス許可が含まれ、すべて CreatedBy:AWSTransform または でタグ付けされたターゲットアカウント内のリソースに制限されます CreatedFor:AWSTransform。
- 移行ウェーブを管理します。
- ネットワーク設定 (Amazon EC2、VPC、Transit Gateway、Direct Connect、ロードバランサー、Network Firewall) を管理します。
- CloudFormation スタックのデプロイを管理します。
- Systems Manager を使用して自動エージェントインストールを実行します。
- 検出リージョンに保存されている情報を使用して、オンプレミスワークロードをターゲット AWS アカウント とリージョンに移行します。
- ターゲットとリージョンでランディングゾーンインフラストラクチャをプロビジョニング AWS アカウント および管理します。これには、以下の項目のアクセス許可が含まれ、CreatedBy:AWSTransform 該当する場合、 でタグ付けされたリソースに限定されます。
- で始まるバケットに対して Amazon S3 バケットオペレーション (作成、読み取り、書き込み、削除) を実行します transform-vmware-landing-zone-。
- ランディングゾーン CloudFormation スタックのスタックデプロイと変更セットを管理します。
- AWS Control Tower オペレーションを実行します。ランディングゾーンの管理、ベースラインの有効化、コントロールの有効化を行うことができます。
- AWS Organizations を管理します。組織単位の作成と管理、アカウントの作成、アカウントの移動を行うことができます。
- Control Tower を使用して AWS サービスコントロールポリシー (SCPs) を管理します。
- AWS Service Catalog プロビジョニングアーティファクトを管理します。

Note

コネクタタイプは、新機能でアクセス許可の変更が必要な場合に更新される場合があります。ターゲットアカウントコネクタタイプの現在のバージョンは 2.0 です。新しいコネクタを作成すると、最新バージョンが使用されます。

コネクタを設定する前に、移行に関連するアカウントロールを理解してください。

アカウント	説明
AWS アカウントを交換する	Transform をセットアップした AWS Organization のメンバーアカウント AWS。これは、AWS 変換ワークスペースが実行される場所です。管理アカウントである必要はありません。
コネクタターゲットアカウント	Transform AWS コネクタが設定されているアカウント。これは移行タイプによって異なります。 • シングルアカウント移行 – ワークロードを移行するアカウントに接続します。コネクタターゲットアカウントとターゲットアカウントは同じです。 • マルチアカウント移行 – 組織管理アカウントまたは委任管理者 (DA) アカウントに接続します。DA アカウントは、組織内の AWS MGN と CloudFormation StackSets の両方の委任管理者として登録する必要があります。AWS Transform は、接続されたアカウントが管理アカウントであるか DA アカウントであるかを確認し、それに応じて動作を調整します。
ターゲットアカウント	ワークロードが移行される AWS アカウント。単一アカウント移行の場合、これはコネクタターゲットアカウントと同じです。マルチアカウント移行では、これらは移行されたワークロードを受け取る個々のメンバーアカウントです。

委任管理者アカウントの使用

マルチアカウント移行の場合、AWS では、組織管理アカウントを直接使用するのではなく、委任管理者 (DA) アカウントを使用することをお勧めします。DA アカウントは、移行オペレーションに必

要なアクセス許可の範囲を制限することで、最小特権の原則に従います。DA アカウントは、AWS 組織内の MGN と CloudFormation StackSets の両方の委任管理者として登録する必要があります。

2 つのオプションの主な違いは次のとおりです。

- 管理アカウント – 組織全体で MGN と CloudFormation StackSets の信頼されたアクセスを有効にできます。を使用して calls CloudFormation StackSets APIs を AWS 変換します CallAs: SELF。
- 委任管理者アカウント – 信頼されたアクセスを直接有効にすることはできません (管理アカウントから行う必要があります)。ただし、MGN ソースサーバー、起動インスタンス、およびメンバーアカウント間での deploy CloudFormation StackSets を管理できます。AWS を使用して calls CloudFormation StackSets APIs を変換します CallAs: DELEGATED_ADMIN。

詳細については、[「MGN ユーザーガイド」の「MGN の委任管理者」](#)を参照してください。

コネクタのセットアップ中に作成された IAM ロール

コネクタのセットアップ中に、AWS Transform はターゲットアカウントに次の IAM ロールを作成します。

- AWSTransform-Connector-role- – AWS 組織の管理アカウントまたは委任管理者アカウントで、ターゲットアカウントコネクタをセットアップしたときに作成されます。このロールにより、AWS Transform はターゲットアカウントに接続し、ユーザーに代わって移行オペレーションを実行できます。

ターゲットアカウントコネクタのセットアップ

Important

コネクタのセットアップ中に、Amazon S3 バケットがターゲットに作成されます AWS アカウント。デフォルトでは、このバケットは HTTPS のみのアクセス (SecureTransport) を強制しません。バケットポリシーに安全なトランスポートを含める場合は、ポリシーを自分で更新する必要があります。詳細については、[Amazon S3のセキュリティのベストプラクティス](#)」を参照してください。

既存のターゲットアカウントコネクタを使用するには

1. ジョブプランペインで、ターゲットアカウントの選択を展開し、コネクタの作成または選択を選択します。
2. コラボレーションタブで、既存のコネクタを選択し、コネクタの使用を選択します。コネクタが使用できない場合、そのバージョンは選択したジョブタイプと互換性がありません。

 Important

AWS 変換リージョンとは異なるターゲットを持つコネクタを指定すると、AWS Transform AWS リージョン はデータを転送します AWS リージョン。

3. [続行] をクリックしてください。

新しいコネクタを作成するには

1. ジョブプランペインで、Connect ターゲットアカウントを展開し、コネクタを作成または選択します。
2. AWS リージョン ターゲットの AWS アカウント と を指定し、次へ を選択します。

 Important

ターゲットが検出 と AWS リージョン 異なる場合 AWS リージョン、AWS Transform はデータを 間で転送します AWS リージョン。

3. 暗号化に Amazon S3 マネージドキーを使用するかどうかを選択します。独自の KMS キーを指定する場合は、デフォルトのキーポリシーを使用するか、許可の低いポリシーを使用できます。KMS キーの作成の詳細については、「AWS Key Management Service デベロッパーガイド」の「[Create a KMS key](#)」を参照してください。

AWS Transform は、アクセス `kms:DescribeKey` 許可を使用してキーが存在することを確認し、Amazon `kms:GenerateDataKey` `kms:Decrypt` S3 バケット内のジョブデータを暗号化および復号します。Amazon S3 詳細については、「[Amazon S3 バケットキーを使用した SSE-KMS のコストの削減](#)」を参照してください。

4. [続行] をクリックしてください。
5. 検証リンクをコピーし、ターゲットの管理者と共有して AWS アカウント、接続リクエストを承認するように依頼します。

6. 管理者がリクエストを承認したら、リストから新しく作成したコネクタを選択し、コネクタの使用を選択します。
7. Send to AWS Transform を選択します。

テンプレートを変更 AWS Transform MGN して起動後のアクションを有効にする場合は、ターゲットコネクタロールに次のアクセス許可を追加します。この JSON ポリシーステートメントは、起動後のアクションロールのiam:PassRoleアクセス許可を付与します。コネクタの作成後、コラボレーションタブでロール名を確認できます。ロールにアクセス許可を追加する方法については、IAM ユーザーガイドの「[ロールのアクセス許可の更新](#)」を参照してください。

```
{
  "Sid": "MGNPostLaunchActions",
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": "arn:aws:iam::target-account-ID:role/service-role/AWSApplicationMigrationLaunchInstanceWithSsmRole"
}
```

サポートされているターゲットリージョン

移行ターゲットリージョンは、ランディングゾーン、ネットワークインフラストラクチャ、サーバーのリホストなど、移行されたリソース AWS リージョン がデプロイされる です。コネクタを作成するときは、ターゲットを指定します AWS リージョン。次のいずれかを使用できます AWS リージョン。

- 米国東部 (バージニア北部)
- 米国東部 (オハイオ)
- 米国西部 (北カリフォルニア)
- 米国西部 (オレゴン)
- アフリカ (ケープタウン)
- アジアパシフィック (香港)
- アジアパシフィック (台北)
- アジアパシフィック (ムンバイ)
- アジアパシフィック (ハイデラバード)

- アジアパシフィック (東京)
- アジアパシフィック (ソウル)
- アジアパシフィック (大阪)
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)
- アジアパシフィック (ジャカルタ)
- アジアパシフィック (メルボルン)
- アジアパシフィック (マレーシア)
- アジアパシフィック (ニュージーランド)
- アジアパシフィック (タイ)
- カナダ (中部)
- カナダ西部 (カルガリー)
- 欧州 (フランクフルト)
- 欧州 (チューリッヒ)
- 欧州 (アイルランド)
- 欧州 (ロンドン)
- 欧州 (パリ)
- 欧州 (ストックホルム)
- 欧州 (ミラノ)
- 欧州 (スペイン)
- イスラエル (テルアビブ)
- メキシコ (中部)
- 南米 (サンパウロ)

 Important

AWS 変換 AWS リージョンとは異なるターゲットを指定すると AWS リージョン、データの一部が間で転送されます AWS リージョン。

サーバーレプリケーションデータは、ソース環境からターゲットアカウントとリージョンに直接送信されることに注意してください。

ランディングゾーンを構築する

AWS Transform は、移行プロジェクトの一環としてラン AWS ディングゾーンを設計およびデプロイするガイドを提供します。ランディングゾーンは、ワークロードが到着する前に、組織の境界、ガバナンスコントロール、アカウント構造を備えたワークロードの基盤として機能するマルチアカウント AWS 環境です。AWS Transform は、移行インベントリとビジネス要件を分析して、組織単位 (OU) とアカウント構造を推奨し、推奨されるサービスコントロールポリシー (SCPs) を適用し、Infrastructure as Code (IaC) を生成および/またはデプロイします。

ランディングゾーンエージェントは、次の 2 つのフェーズを順を追って説明します。

- 基盤のセットアップ – コアランディングゾーン構造、AWS Control Tower、基盤 OUs、コアアカウントを確立します。
- ワークロードアカウント設計 – 移行の波、ビジネスユニット、環境分離要件に基づいてワークロード OUs とアカウントを設計および作成します。

AWS Transform は、グリーンフィールド環境 (既存のランディングゾーンなし) とブラウンフィールド環境 (既存の OUs とアカウントがデプロイ済み) の両方をサポートしています。ブラウンフィールドシナリオでは、AWS Transform は既存の組織構造を検出し、AWS ベストプラクティスとのギャップを埋めるために必要な変更のみを推奨します。

コネクタのセットアップ

ランディングゾーンエージェントには、組織管理アカウントにリソースをプロビジョニングするためのターゲット AWS アカウント コネクタが必要です。コネクタには、次のアクセス許可があります。

- [AWS Control Tower](#) のセットアップ
- [組織単位とアカウント](#)を作成する
- [サービスコントロールポリシー \(SCPs\)](#)を設定する

コネクタリクエストを承認すると、AWS 変換アクセス許可が付与されます。

- ターゲットとリージョンでランディングゾーンインフラストラクチャをプロビジョニング AWS アカウント および管理します。これには、以下の項目のアクセス許可が含まれ、ATWorkspace: {workspace-id} 該当する場合、CreatedBy:AWSTransform および でタグ付けされたリソースに制限されます。

- で始まるバケットの S3 バケットオペレーション (作成、読み取り、書き込み、削除)
transform-vmware-landing-zone-
- CloudFormation ランディングゾーンスタックのスタックデプロイと変更セット管理
- AWS Control Tower オペレーション (ランディングゾーンの管理、ベースラインとコントロールの有効化)
- [AWS 組織](#)の管理 (組織単位の作成と管理、アカウントの作成、アカウントの移動)
- AWS Control Tower によるサービスコントロールポリシー (SCP) 管理
- [AWS Service Catalog](#) プロビジョニングアーティファクト管理

コネクタを作成するときは、ターゲットを指定します AWS リージョン。このリージョンは、ホーム Control Tower リージョンと同じである必要があります。Control Tower リージョンの詳細については、[AWS 「Control Tower AWS リージョン の操作」方法](#)を参照してください。

ランディングゾーンのセットアップの開始時に、AWS Transform はコネクタ設定を取得し、確認のために AWS Organization 管理アカウント ID とターゲットリージョンを表示します。詳細については、「[Connector](#)」を参照してください。

Important

IAM アイデンティティセンターリージョンの依存関係 – AWS 変換には AWS IAM アイデンティティセンター (IAM アイデンティティセンター) が必要です。つまり、コネクタリージョンは AWS Control Tower ホームリージョンと IAM アイデンティティセンターリージョンの両方と一致する必要があります。IAM Identity Center が組織で既に設定されている場合、AWS コネクタが別のリージョンをターゲットとしている場合、Control Tower の初期化は失敗します。詳細については、「Control Tower ユーザーガイド」の「[IAM Identity Center のお客様向けの考慮事項 AWS](#)」を参照してください。

基盤のセットアップ

基盤設定フェーズでは、[AWS Control Tower](#) を使用してコアランディングゾーンインフラストラクチャを確立します。AWS Control Tower がランディングゾーンを設定すると、管理アカウントに一連のマネージドリソースが自動的にプロビジョニングされ、AWS 組織全体のガバナンス基盤を形成します。

- ルート – ランディングゾーン内のすべての OUs を含む最上位の親。

- セキュリティ OU – Control Tower によって自動的に作成されます。ログアーカイブアカウント (組織全体のすべての AWS API アクティビティとリソースの変更に対する一元管理されたイミュータブルなログ記録) と監査アカウント (セキュリティとコンプライアンスのレビューのためのすべてのアカウントへの読み取り専用アクセス) の 2 つの共有アカウントが含まれます。これらのアカウントの名前を変更したり、初期設定後に置き換えたりすることはできません。
- 必須コントロール (ガードレール) – Control Tower は、予防コントロールと検出コントロールを組織全体に自動的に適用して、ベースラインガバナンスポリシーを適用します。これらを無効にすることはできません。
- IAM Identity Center ディレクトリ – Control Tower は、ランディングゾーンユーザーの事前設定されたグループとシングルサインオンアクセスを持つクラウドネイティブディレクトリを作成します。詳細については、[AWS「IAM Identity Center」](#)を参照してください。

Control Tower は [CloudFormation StackSets](#) を使用して、組織内のすべてのアカウントとリージョンにこれらのリソースを一貫してデプロイおよび管理します。ランディングゾーンが不明な状態になる可能性があるため、サポートされている方法以外で Control Tower マネージドリソースを変更または削除しないでください。

アカウントの E メール規則

AWS では、アカウントごとに一意の E メールアドレスが必要です。これらの E メールは、アカウントに関する重要な通知を受け取ります。AWS Transform は、と アドレス指定を使用して、単一のメールボックスから一意のアカウント E メールを生成します。

形式: prefix+account-name@domain

プレフィックス (などaws-admin) とドメイン (などacme.com) を指定すると、AWS 変換によってすべてのアカウント E メールが自動的に取得されます。例えば、次のようになります。

- 監査アカウント: aws-admin+audit@acme.com
- ログアーカイブアカウント: aws-admin+log-archive@acme.com
- サンドボックスアカウント: aws-admin+sandbox@acme.com

ブラウンフィールドシナリオでは、AWS Transform は既存のアカウント E メールを検査して、既に使用されているプラスアドレス規則を推測し、同じパターンを続行することを提案します。

推奨される基盤構造

AWS ベストプラクティスに基づいて、AWS Transform は次の基盤 OU 構造を推奨します。作成前にカスタマイズできます。

OU	目的	アカウント
セキュリティ	一元化された監査ログ記録とモニタリング。これらのサービスを専用アカウントで分離することは、監査証跡をワークロードチームと分離するのに役立ちます。	監査、ログアーカイブ
インフラストラクチャ	共有ネットワーキング (Transit Gateway、VPN)、DNS、および共通サービス。重複を減らし、ネットワークチームが 1 か所で接続を管理することができるように、これらを一元化することをお勧めします。	なし (空で作成)
サンドボックス	支出制限とアクセス制限に関するデベロッパー実験。本番環境のリソースを危険にさらすことなく、デベロッパーに実験するスペースを与えることをお勧めします。	サンドボックス
ワークロード	Production、Non-Production、およびオプションで規制されたサブ OUs。ワークロードアカウントは、移行要件に基づいて次のフェーズで設計されています。	なし (空で作成)

Note

[監査アカウントとログアーカイブアカウント](#)を含むセキュリティ OU は、Control Tower 基盤設定の一部として作成されます。インフラストラクチャ、サンドボックス、ワークロード OUs は、構造を確認した後に個別に作成されます。

ブラウンフィールドシナリオでは、AWS Transform は既存の基盤をこの推奨構造と比較し、ギャップのみをレポートします。例：「基盤にはセキュリティとインフラストラクチャの OUs」

サービスコントロールポリシー (SCP)

SCP は、[組織内のすべてのアカウントに最大アクセス許可を設定する組織レベルのアクセス許可ガードレール](#)です。AWS アクセスを許可しません。代わりに、アカウント管理者であっても、アカウント内の誰も越えることができない境界を定義します。

Control Tower のデプロイの一環として、ベースラインガードレールが自動的に適用されます。AWS Transform は、組織の体制を強化するために設計された追加の SCPs も推奨します。これらは、実行可能な最小ランディングゾーンの AWS ベストプラクティスに基づいています。

SCP は、インフラストラクチャ、サンドボックス、ワークロード OUs に適用できます。Security OU は Control Tower によって管理され、このツールを通じて SCPs によってターゲットにすることはできません。

Important

Security OU は、Control Tower によって管理される基盤 OU です。ランディングゾーンエージェントを介してアカウント、SCP、またはリソースを追加することはできません。

ブラウフィールドシナリオでは、AWS Transform はどの SCPs がすでに適用されているかをチェックし、ギャップを埋める SCP のみを推奨します。

Foundation のデプロイ

基盤設計が完了したら、デプロイ方法を選択します。

- Deploy for me – AWS Transform は基盤 OUs、アカウント、SCP を Organization にデプロイします AWS。
- 自分でデプロイする – AWS 変換は、任意の形式でダウンロードするための Infrastructure as Code (IaC) アーティファクトを生成します (「」を参照)[the section called “IaC 形式”](#)。
- ワークロードアカウントを最初に設計する – デプロイをスキップし、ワークロードアカウント設計フェーズに進みます。すべては後でまとめてデプロイできます。

Control Tower の初期化

AWS Transform は、組織内で AWS Control Tower がまだ初期化されていないことを検出した場合、AWS 変換コンソールページへのリンクをユーザーに提供します。リンクで オペレーションを生成すると、Control Tower をブートストラップする CloudFormation スタックが作成されます。このプロ

セスでは、ターゲットリージョンの CloudFormation コンソールにこのスタックが作成されます。スタックの作成が完了すると、AWS Transform はデプロイを続行します。

ワークロードアカウント設計

ワークロードアカウント設計フェーズでは、AWS Transform は、移行インベントリ、ビジネス要件、環境分離設定に基づいて、アプリケーションワークロードの OU とアカウント構造を設計します。

移行計画コンテキスト

AWS Transform は、ウェーブプラン、server-to-applicationマッピング、共有コンテキストなど、移行計画フェーズからデータを取得します。移行計画データが利用可能な場合、AWS Transform は概要を表示し、確認または調整するよう求めます。移行計画データが利用できない場合、AWS Transform は検出に直接質問をします。

発見

AWS Transform は、ワークロードの要件を理解するために質問します。質問はスキップできます。トピックは以下のとおりです。

- を使用するビジネスユニットまたはチームの数 AWS
- 業界および該当するフレームワーク (HIPAA、PCI-DSS、SOC2、FedRAMP)
- ワークロードが機密データ (PII、PHI、財務) を処理するかどうか
- 環境分離設定 (開発dev/test/staging/個別のアカウントまたは共有として生成)
- ワークロードの分離要件
- ビジネスアプリケーションとその目的
- アプリケーションへのサーバーグループ化
- コスト追跡と配分のニーズ (ビジネスユニット、プロジェクト、環境別)
- 今後 12~24 か月に予想される増加
- アカウント戦略の設定 (アカウント、グループ化、または環境ベースごとに 1 つのアプリ)

提案されたワークロード構造

回答と移行計画データに基づいて、AWS Transform はワークロード OU の下に OU とアカウント構造を提案します。この提案には、各設計決定の背後にある推論が含まれています。

AWS 変換は、次の設計原則に従います。

- 移行ウェーブ内のすべてのサーバーは同じアカウントに送られます。ウェーブをアカウント間で分割することはできません。これは、ウェーブ実行中のリホスト制限です。
- 分離された環境をリクエストすると、AWS Transform はワークロード/本番稼働用およびワークロード/非本番稼働用サブ OUs を作成します。
- 該当するフレームワークが特定された場合、AWS Transform はワークロード/規制対象およびワークロード/標準サブ OUs を作成します。
- 複数のビジネスユニットが異なるガバナンスを必要とする場合、AWS Transform はワークロードの下に business-unit-specific OUs を作成します。
- 重要または機密データアプリケーションは、アカウントごとに 1 つのアプリケーションを取得します。この場合、ウェーブプランを反復するように求められることがあります。
- 依存関係を共有する緊密に結合されたアプリケーションは、1 つのアカウントにグループ化されます。

提案された各アカウントには、名前、目的、ターゲット OU、ビジネスユニットが含まれます。AWS Transform は、使用されている命名規則を示します (例: <business-unit>-<environment>-<workload>)。

AWS Transform が変更を適用する前に、提案された構造を確認して変更できます。適用後、反復して、満足するまで追加の変更を加えることができます。

ワークロード SCP 設定

ワークロード構造が作成されると、AWS Transform は使用可能な SCPs を提示し、ワークロード OUs。適用 SCPs と OUs。AWS Transform が SCPs し、更新された組織ツリーを SCP 概要テーブルとともに表示します。

ワークロードのデプロイ

ワークロード設計が完了したら、デプロイ方法を選択します。

- Deploy for me – AWS Transform はワークロード OUs、アカウント、SCPs を Organization にデプロイします AWS。
- 自分でデプロイする – AWS 変換は IaC アーティファクトを生成して任意の形式でダウンロードします (「」を参照 [the section called “IaC 形式”](#))。

laC 形式

自己デプロイを選択すると、AWS Transform は次の形式で Infrastructure as Code アーティファクトを生成します。

- [AWS Cloud Development Kit \(AWS CDK\)](#) – プログラムによるインフラストラクチャデプロイ用の TypeScript プロジェクト。
- HashiCorp Terraform – ランディングゾーンリソースを管理するための HashiCorp 設定言語 (HCL) テンプレートを生成します。
- Landing Zone Accelerator (LZA) – LZA ユニバーサル設定バージョン 1.1.0 に基づく設定 YAML ファイル。これらのエンタープライズ対応テンプレートは、の Landing Zone Accelerator と連携して、マルチアカウント AWS 環境を確立 AWS します。生成されたファイルには、AWS ベストプラクティスに沿ったガバナンス、組織構造、ネットワークのための事前設定が含まれています。詳細については、[「LZA ユニバーサル設定」](#)を参照してください。

Note

Landing Zone Accelerator (LZA) パイプラインを介してデプロイする場合、AWS 変換アカウントと LZA のインストールは同じ AWS Organization に存在する必要があります。AWS Transform と LZA で使用される Organizations IDs が一致しない場合、デプロイは失敗します。Organizations を使用して LZA インストールを設定する方法については、[AWS 「Organizations ベースのインストール」](#)を参照してください。

形式を選択すると、AWS Transform はアーティファクトを生成し、ダウンロードできるようにします。

ダウンロードしたファイルが破損または改ざんされていないことを確認するには、チェックサムを生成してダウンロードし、以下を使用してローカルに生成されたハッシュと比較します。

```
openssl dgst -sha256 -binary <file.zip> | base64
```

デプロイの承認プロセス

ランディングゾーンのデプロイリクエストには、実行前に明示的な承認が必要です。デプロイリクエストを送信すると、AWS Transform Approvals タブを介して承認済承認者に自動的にルーティングされます。

承認者は CloudFormation テンプレートとランディングゾーンの設定を確認します。AWS Transform の管理者ロールを持つユーザーのみがデプロイリクエストを承認できます。各送信は新しいレビューサイクルをトリガーし、デプロイは確認を受け取った後にのみ続行されます。

承認者がリクエストを拒否した場合は、直接連絡して必要な変更について話し合います。システムは、監査目的でのすべての承認決定を追跡し、デプロイ履歴を維持します。

ランディングゾーンリソースにタグを付ける

AWS 変換は、追跡のために、生成されたすべてのリソースに定義と実行 IDs "CreatedBy": "AWSTransform"とともに自動的にタグ付けします。

自動タグ

すべてのランディングゾーンリソースには、次のタグが付けられます。

- CreatedBy – AWSTransform
- ATWorkspace – ワークスペース識別子

Note

移行が AWS Migration Acceleration Program (MAP 2.0) の一部である場合は、必要な MAP タグとして Key: map-migrated Value: migMPE_ID (MPE_ID は移行ポートフォリオ評価識別子) を含めることができます。MAP タグは、コネクタのセットアップフェーズ中にリクエストされます。AWS Transform は、ランディングゾーンのデプロイ中にこれらのタグを適用します。

変更の取り消し

削除できるのは、デプロイされていない要素のみです。OU またはアカウントがデプロイされると、ランディングゾーンエージェントを介して削除することはできません。

要素を削除する場合、順序は重要です。親の前に子を削除する必要があります。

1. まず (E メールで) アカウントを削除します。
2. OUs から SCPs を削除します。
3. 子 OUs の削除 — アカウントまたはネストされた OU がまだある場合 OUs を削除することはできません。

関連リソース

- [the section called “ターゲットの AWS アカウントとリージョンを接続する”](#)
- [the section called “ネットワークを に移行する AWS”](#)
- [AWS Control Tower ユーザーガイド](#)
- [AWS Organizations ユーザーガイド](#)
- [AWS IAM Identity Center ユーザーガイド](#)
- [CloudFormation ユーザーガイド](#)
- [でのランディングゾーンアクセラレーター AWS](#)
- [AWS Well-Architected フレームワーク](#)
- [AWS 規範ガイダンス: ランディングゾーンの構築](#)

ネットワークを に移行する AWS

AWS Transform を使用すると、ネットワークを に移行できます AWS。AWS Transform は、必要に応じてソース環境設定を VPCs、サブネット、セキュリティグループ、NAT ゲートウェイ、トランジットゲートウェイ、Elastic IPs、ルート、ルートテーブルなどの AWS 同等のネットワークリソースに変換します。デプロイ前に、生成されたネットワーク設定を確認および変更できます。ネットワーク接続の AWS 変換と分析を使用して設定をデプロイできます。または、自己デプロイを選択し、Infrastructure as Code (IaC) を任意の形式で受け取ることもできます。、Landing Zone Accelerator (LZA) AWS Cloud Development Kit (AWS CDK)、または HashiCorp Terraform。

ネットワークを移行するには、次の手順に従います。

1. ソースネットワークファイルをアップロードします。
2. 追加の設定ファイルをアップロードします (オプション、RVTools 環境の場合)。
3. ネットワークトポロジを選択します。
4. セキュリティグループのマッピング戦略を選択します。
5. ネットワークを確認して最適化します。
6. ネットワーク図を生成します (オプション)。
7. リソースのタグ付けを設定します。
8. ネットワークをデプロイします。

Note

マルチアカウントデプロイでは、ネットワーク移行を開始する前に、クロスアカウント IAM ロールと AWS Organizations の信頼されたアクセスを設定する必要があります。移行タイプの詳細については、「」を参照してください [the section called “ステップ 1: 移行タイプの選択”](#)。

ステップ 1: ソースネットワークマッピング

ネットワークマッピングプロセスでは、ソース環境から設定ファイルをアップロードする必要があります。選択するツールは、ソースネットワークタイプによって異なります。

- Software Defined Networks (SDN): VMware NSX ネットワーク仮想化の場合は Import/Export、Cisco Application Centric Infrastructure の場合は Cisco ACI 設定。
- VMware vSphere ネットワーク: [RVTools](#)。RVTools ファイルを使用する場合、AWS Transform は Amazon VPC 設定のみを生成します。セキュリティグループ設定には、ファイアウォールまたはソフトウェア定義のネットワークファイルからの追加の入力が必要です。追加のファイルからセキュリティグループを生成する方法の詳細については、「[追加の設定ファイル](#)」を参照してください。
- ファイアウォール設定データに基づくネットワーク: Palo Alto Networks Firewall、Fortinet FortiGate Firewall、または Cisco ACI からファイルをエクスポートします。サポートされているバージョンと抽出手順の詳細については、「[設定ファイルの抽出](#)」を参照してください。
- VMware ワークロードと非 VMware ワークロードの両方を実行するハイブリッドネットワーク: [AWS 検出ツールまたは modelizeIT を変換](#)します。modelizeIT
- その他のファイルタイプ: 設定ファイルが上記のサポートされている形式のいずれでもない場合、ファイルはサポートされている形式に自動的に変換されます。この変換には、ファイルサイズと複雑さに基づいて最大 2 時間かかる場合があります。

Note

サポートされているソースネットワークファイルの最大サイズは 70 MB です。

⚠ Warning

<https://www.dell.com/en-us/shop/vmware/sl/rvtools> の公式の Dell サイトからのみ RVTools をダウンロードします。非公式のソースから RVTools をダウンロードしないでください。

各ソースネットワークセグメントは、独自の VPC にマッピングされます。ネットワークセグメンテーションはソースタイプによって異なります。

- vNetwork: AWS VMs vSwitch と仮想 LAN (VLAN) で変換します。VLANsは複数の vSwitchesの下に表示されます (VLAN 0 を除く)。
- NSX ネットワーク: AWS Tier-1 ルーターに基づいてネットワークをセグメント化し、ルーターをグループ化してセグメントを収集します。

ステップ 2: 追加の設定ファイル

RVTools ソース環境では、オプションで追加の設定ファイルをアップロードして、セキュリティグループの生成を有効にすることができます。追加の設定ファイルをアップロードしない場合、RVTools ベースの移行用のセキュリティグループは生成されません。

AWS Transform では、次の追加の設定ファイルタイプがサポートされています。1 つのプラットフォームからアップロードできる設定ファイルは 1 つだけです。

- Cisco Application Centric Infrastructure (ACI) は、ネットワークポリシー設定を提供します。
- Palo Alto Networks はファイアウォールセキュリティポリシーを提供します。
- Fortinet FortiGate は、ファイアウォールセキュリティポリシーを提供します。

ファイアウォールまたは Cisco ACI ファイルをアップロードすると、AWS Transform はネットワークインフラストラクチャとセキュリティグループを生成します。RVTools ファイルを単独でアップロードすると、AWS Transform はネットワークインフラストラクチャのみを生成します。

サポートされているバージョンと抽出手順の詳細については、「[設定ファイルの抽出](#)」を参照してください。

ステップ 3: ネットワークトポロジ

ネットワーク定義ステップでは、ネットワークトポロジを選択します。分離された VPCs トポロジまたは Hub トポロジとスポークトポロジを選択できます。

分離された VPC

デプロイされる内容

分離された VPCs は、内で個別のユニットとして動作する独立したネットワーク環境です。AWS VPCs は完全に分離され、それらの間に組み込みの通信パスはありません。この分離により、最高レベルのネットワーク境界保護が提供されます。

AWS Transform は、次のリソースを作成します。

- 検出されたソースネットワークセグメントごとに専用の VPC。
- ソースネットワーク設定に基づくプライベートサブネット。
- セキュリティグループ (ファイアウォールまたは SDN 設定ファイルを指定した場合)。

セットアップを完了する

AWS Transform はネットワークインフラストラクチャをデプロイしますが、インターネットアクセスと VPC 間接続を残すため、組織の要件を満たす設定を選択できます。

分離された VPC のインターネットアクセスを有効にするには、次の手順を実行します。

1. [インターネットゲートウェイ](#)を作成し、VPC にアタッチします。
2. インターネットアクセスが必要な各アベイラビリティゾーンにパブリックサブネットを作成します。インターネットゲートウェイ 0.0.0.0/0 を指すルートを追加します。サブネット設定の詳細については、[「VPC のサブネット」](#)を参照してください。
3. パブリックサブネットに [NAT ゲートウェイを作成します](#) (高可用性のために AZ ごとに 1 つ)。NAT ゲートウェイごとに Elastic IP を割り当てます。
4. プライベートサブネット [ルートテーブル](#)を更新する — 同じ AZ 内の NAT ゲートウェイ 0.0.0.0/0 を指すルートを追加します。
5. [セキュリティグループルール](#)を確認する — アウトバウンドルールで、ワークロードに必要なトラフィック (HTTPS、DNS など) が許可されていることを確認します。

VPC-to-VPC [ピアリング](#)または [Transit Gateway](#)を設定し、各 VPC のルートテーブルを更新して、トラフィックをピアリング接続または TGW アタッチメントにルーティングします。

ハブとスポーク

このモデルでは、[AWS Transit Gateway](#) は複数のワークロード VPCs (スポーク) を接続する中央ハブとして機能します。

デプロイされる内容

AWS Transform は、次のリソースを作成します。

- スポーク VPCs: 検出されたソースネットワークセグメントごとに 1 つの VPC、プライベートサブネットと Transit Gateway アタッチメント。
- 検査 VPC: トラフィック検査のためにファイアウォールアプライアンスをホストします。すべてのクロス VPC トラフィックは、この VPC を介してルーティングされます。Transit Gateway アタッチメントは、[アプライアンスモード](#)を使用します。これは、トラフィックが接続の両方向に同じアプライアンスを対称的に流れるようにする設定です。
- インバウンド VPC: パブリックインターネット (南北インバウンド) からネットワークに入るトラフィックを処理します。複数のアベイラビリティゾーンにまたがる[インターネットゲートウェイ](#)とパブリックサブネットが含まれます。
- アウトバウンド VPC: ネットワークからパブリックインターネット (南北アウトバウンド) へのトラフィックを処理します。インターネットゲートウェイ、高可用性のために各アベイラビリティゾーンに [Elastic IP アドレス](#)を持つ [NAT ゲートウェイ](#)、および Transit Gateway アタッチメントのプライベートサブネットが含まれます。
- Transit Gateway ルートテーブル: 2 つのルートテーブルがインスペクション VPC を介してトラフィックを誘導します。未検査テーブルは、スポーク VPCs、インバウンド VPC、アウトバウンド VPC に関連付けられています。すべてのトラフィック (0.0.0.0/0) を検査 VPC アタッチメントにルーティングし、デフォルトの関連付けルートテーブルです。Inspected テーブルは Inspection VPC に関連付けられています。これにはすべてのスポーク VPCs、デフォルトの伝播ルートテーブルです。

マルチアカウントデプロイの場合、Transit Gateway は [AWS Resource Access Manager \(RAM\)](#) を介してアカウント間で共有されます。

トラフィックフロー

すべてのクロス VPC トラフィックは次のパスに従います。

1. スポーク VPC からのトラフィックは Transit Gateway (デフォルトルート 0.0.0.0/0) に送信されます。

2. 未検査ルートテーブルは、トラフィックを検査 VPC にルーティングします。
3. 検査 VPC のファイアウォールはトラフィックを検査し、トランジットゲートウェイに転送します。
4. Inspected ルートテーブルは、伝播されたルートを使用してトラフィックを送信先スポーク VPC にルーティングします。

アウトバウンドインターネットトラフィックの場合、検査済みルートテーブルはトラフィックをアウトバウンド VPC にルーティングします。NAT ゲートウェイは、トラフィックがインターネットゲートウェイに転送される前にプライベート IP アドレスを変換します。アウトバウンド VPC パブリックルートテーブルには、Transit Gateway に戻る各スポーク VPC クラスレスドメイン間ルーティング (CIDR) 範囲の特定のルートが含まれます。これらのルートにより、リターントラフィックが正しいスポーク VPC に到達できるようになります。

インバウンドインターネットトラフィックは、インバウンド VPC のインターネットゲートウェイを経由して入り、同じ検査パスに従ってスポーク VPCs。

セットアップを完了する

AWS Transform はネットワークインフラストラクチャをデプロイしますが、ファイアウォール設定とインバウンドサービスの設定はそのままにしておくため、組織の要件を満たすセキュリティアプライアンスとポリシーを選択できます。

 Note

デフォルトでは、クロス VPC トラフィックは検査なしで検査 VPC を通過します。トラフィック検査を有効にするには、ファイアウォールをデプロイする必要があります。

ファイアウォールをデプロイする: ファイアウォールエンドポイントの検査 VPC に追加のサブネットを作成します。AWS Transform は Transit Gateway アタッチメントのサブネットのみを作成します。TGW アタッチメントサブネットからファイアウォールエンドポイントにトラフィックをルーティングし、ファイアウォールサブネットから Transit Gateway にトラフィックをルーティングします。[AWS Network Firewall](#) またはサードパーティーアプライアンスをデプロイできます。Transit Gateway を使用したファイアウォールのデプロイの詳細については、「[Creating a firewall with a Transit Gateway](#)」を参照してください。

接続の確認: ファイアウォールをデプロイしたら、スポーク VPC インスタンス (など) からのアウトバウンドインターネットアクセスをテストします `curl https://aws.amazon.com`。 [Reachability Analyzer](#) を使用して、接続の問題をトラブルシューティングできます。

インバウンドサービスを設定する: パブリック向けサービスをホストするには、 [Application Load Balancer](#) または [Network Load Balancer](#) をインバウンド VPC パブリックサブネットにデプロイします。 Transit Gateway を介してスポーク VPCs 内のインスタンスを指すターゲットグループを設定し、 Inspected ルートテーブルにインバウンド VPC へのリターンルートがあることを確認します。

VPCs 間の通信をきめ細かく制御する場合は、分離 VPCs オプションを選択し、生成されたネットワークを変更して、必要な特定の通信パスを作成します。

ステップ 4: セキュリティグループのマッピング

ソースセキュリティポリシーを AWS セキュリティグループに変換する方法を選択します。 AWS Transform は、ソース環境設定に基づいてセキュリティグループを作成します。セキュリティポリシー、セキュリティポリシールール、ゲートウェイポリシー、ゲートウェイポリシールールは、セキュリティグループに変換されます。

Important

AWS Transform は、ソース環境に合わせてベストエフォートベースでセキュリティグループを作成します。生成されたセキュリティグループを確認して変更し、会社のニーズとセキュリティポリシーを満たしていることを確認します。

セキュリティグループの参照

セキュリティグループが生成されると、AWS Transform はサポートされているセキュリティグループ参照を使用します。セキュリティグループ参照は、特定の IP アドレス範囲 (CIDR ブロック) ではなく、別のセキュリティグループ ID に基づいてセキュリティルールを設定します。このアプローチは、より柔軟で保守可能なセキュリティ設定を提供します。

セキュリティグループルールは、同じ VPC 内、または同じリージョン内に接続されている VPC 内の他のセキュリティグループのみを参照できます。クロスアカウント参照もサポートされています。接続されていない VPC またはリージョン間でセキュリティグループを参照することはできません。接続された VPCs、クロス VPC セキュリティグループ参照をサポートするのはインバウンドルールのみです。アウトバウンドルールは CIDR ベースのルールを使用する必要があります。AWS

Transform がセキュリティグループルールを作成する方法は、選択したネットワークポロジによって異なります。

- ハブとスポーク: Transit Gateway は VPCs 間のネットワーク接続を提供します。AWS Transform は、VPC 内およびクロス VPC/クロスアカウント進入ルールの両方に参照を使用します。クロス VPC/クロスアカウント出力 (アウトバウンド) ルールは、CIDR ベースのルールを使用します。
- 分離 VPCs: VPCs 間にネットワーク接続がありません。AWS Transform は VPC 内ルールの参照のみを使用します。すべてのクロス VPC ルールとクロスアカウントルールは CIDR ベースのルールを使用します。

CIDR ベースのルールは、ソース設定が対称でない場合にも使用されます。

次のいずれかのセキュリティグループマッピング戦略を選択します。

- MAP: セキュリティルールをソース環境から AWS セキュリティグループとルールに変換します。このオプションは、静的 IP アドレス指定を使用する移行に使用します。
- MAP_DHCP (DHCP サポートで翻訳): ソース環境のセキュリティルールを DHCP 互換で翻訳します。DHCP は、サブネットの CIDR 範囲から IP アドレスを動的に割り当てます。その結果、VPC 間の出力ルールは、完全な送信先サブネット CIDR と一致するように拡張されます。CIDR を絞り込むと、その範囲外の DHCP 割り当て IPs がブロックされます。移行後にこれらのルールを確認してください。

このオプションは、クロス VPC Transit Gateway 通信での DHCP サポートに使用します。また、静的 IPs でも動作しますが、MAP よりも広範なルールを生成する可能性があります。

- SKIP: セキュリティルールを翻訳しません。移行後に AWS セキュリティグループを手動で設定します。静的 IP 環境と DHCP 環境の両方で動作します。追加の設定ファイルがない RVTools ソース環境の場合、AWS Transform は自動的に SKIP を使用します。

Note

マッピング戦略によって、IP 割り当てオプションが決まります。MAP は静的 IP のみをサポートします。MAP_DHCP と SKIP は、静的と DHCP の両方をサポートします。

IP 移行アプローチ

移行には 2 つのネットワーク設定があります。

ネットワーク範囲の選択

- 既存の範囲を保持 (IP アドレス範囲の保持): 移行中は元の IP アドレス範囲を保持します。特にハードコードされた IP 依存関係や既存のファイアウォールルールを持つレガシーアプリケーションでは、変更 AWS なしでアプリケーションを に移動する (lift-and-shift 移行に最適です)。
- 新しい IP 範囲の更新 (CIDR 更新): 移行中に各 VPC CIDR 範囲を変更でき、AWS Transform はサブネット、ルートテーブル、セキュリティグループに変更を自動的に伝達します。

IP アドレスの割り当て

- 固定 IP アドレス (静的): AWS 変換は CIDR に基づいて静的 IPs を割り当てます。これは、予測可能なネットワーク動作、DNS 管理、または IP ベースのアクセスコントロールを必要とするアプリケーションに最適です。IPs、インスタンスにアタッチされた仮想ネットワークカードである Elastic Network Interface (ENIs) を介してインスタンスの再起動全体で保持されます。
- 動的 IP 割り当て (AWS DHCP): インスタンスの起動時にサブネットプールから IPs を自動的に割り当てます。クラウドおよび Auto Scaling ワークロードで実行されるように設計されたアプリケーションに最適です。運用上のオーバーヘッドを削減しますが、アプリケーションは DNS またはサービス検出を使用する必要があります。

いずれかの範囲選択をいずれかの IP 割り当て方法と組み合わせることができます。

Note

IP アドレス割り当て戦略はウェーブレベルで設定されます。ウェーブファイルをカスタマイズすることで、特定のサーバーにさまざまな戦略を割り当てることができます。たとえば、ウェーブに静的 IP アドレスアプローチを選択し、特定のサーバーに動的アプローチを割り当てている場合は、「MGN ユーザーガイド」の「[設定の編集](#)」で説明 [RESET_VALUE] されているように を使用します。

ステップ 5: ネットワークを確認して最適化する

AWS Transform がターゲットネットワーク設定を生成したら、AWS インフラストラクチャに集約されたオンプレミスネットワークセグメントを確認できます。ビジュアルインターフェイスを使用してネットワークを確認し、チャットインターフェイスを使用して変更を行い、ガイド付きレコメンデーションを受け取ります。AWS Transform はカスケード影響分析を実行し、必要な変更を実装してネットワークの一貫性とベストプラクティスへの準拠を維持します。変換 AWS にネットワークを

分析して最適化を提案するように依頼することもできます。[「ガイド付きレコメンデーション」](#)を参照してください。

ターゲットアカウントの既存の VPCs

ターゲットアカウントに以前の移行フェーズまたは並列インフラストラクチャプロジェクトからの VPCs がすでに含まれている場合、AWS Transform はそれらを自動的に検出し、レビュープロセス中にマッピングされた VPCs とともに表示します。マルチアカウント移行の場合、AWS Transform は AWS Organization 内のすべてのアカウントで既存の VPCs を検出します。

この可視性は、計画されたネットワークが既存のインフラストラクチャにどのように関連しているかを理解し、潜在的な CIDR の競合を特定し、デプロイ前に情報に基づいた意思決定を行うのに役立ちます。

Note

AWS Transform は既存の VPCs のみを検出します (サブネットやその他のリソースは検出しません)。検出は読み取り専用 — AWS 変換は既存の VPCs を変更しません。

ネットワークの最適化

Note

これらのオペレーションはワークロード VPCs にのみ適用されます。ハブおよびスポークトポロジ (検査、インバウンド、アウトバウンド) VPCs では、IP アドレスの変更のみがサポートされます。

削除、マージ、分割の各オペレーションを元に戻すことはできません。これらの変更を適用する前に、設定を注意深く確認してください。

VPCs では、次のオペレーションを使用できます。

- 削除: 設定から VPC を完全に削除します。これは、移行すべきではない古いネットワークセグメントに使用します AWS。
- 除外: 段階的な移行戦略のために、移行から VPC を一時的に削除します。除外 VPCs はデプロイされませんが、後で再含めることができます。
- 含める: 以前に除外した VPC を移行に追加します。

- マージ: 2 つの VPCs に結合します。最初の VPC はアイデンティティを保持し、2 番目の VPC のすべてのサブネットを吸収します。セキュリティグループはマージされた VPC に移動され、それに応じて関連付けが再構築されます。最初の VPC の CIDR は、元の CIDR の両方を含む最小 CIDRs 範囲に拡張され、ルーティングは自動的に更新されます。2 番目の VPC は設定から削除されます。

マージ要件:

- サブネット CIDRs は、2 つの VPCs 間で重複してはいけません。
- マージされた CIDR は /16 を超えることはできません。
- マルチアカウントデプロイの場合、両方の VPCs を同じアカウントに割り当てる必要があります。
- IP アドレスの変更: プレフィックスの長さを同じままに VPC CIDR の基本 IP アドレスを変更します。AWS Transform は、すべてのサブネット CIDRs で自動的に変換します。たとえば、VPC をからに変更 10.0.0.0/16 すると 10.20.0.0/16、サブネットが から 10.0.1.0/24 に移行されます 10.20.1.0/24。

古い VPC CIDR と正確に一致するセキュリティグループルールは自動的に更新されます。部分的に重複するルールや古い CIDR と一致しないルールは変更されません。変更後にこれらのルールを確認してください。

- 名前の変更: コスト配分、コンプライアンス追跡、運用標準に関する組織の命名規則に合わせて VPC の名前を変更します。
- サイズ変更: VPC CIDR のプレフィックスの長さを変更して、IP アドレス範囲を拡大または縮小します。
 - プレフィックスの長さが短くなる (IPs の数が増える、たとえば /20 から /16): サブネットはより大きな範囲内に収まります。サブネットの変更は必要ありません。
 - プレフィックスの長さの増加 (/16 から /20 など、IPs が少ない): 新しい範囲外のサブネットは、まずサブネットのサイズ変更オペレーションを使用してサイズ変更する必要があります。

古い VPC CIDR と正確に一致するセキュリティグループルールは自動的に更新されます。部分的に重複するルールや古い CIDR と一致しないルールは変更されません。サイズ変更後にこれらのルールを確認してください。

サイズ変更の要件:

- 新しい CIDR は /16 から /28 の間である必要があります。
- 新しい CIDR は、ネットワーク内の他の VPCs (ハブおよびスポークトポロジ) と重複してはいけません。

- CIDR を減らす場合、既存のサブネットはすべて新しい CIDR 内に収まる必要があります。必要に応じて、最初にサブネットのサイズを変更します。
- 分割: 指定した CIDR 境界に基づいて VPC を 2 つの VPCs に分割します。サブネットは、CIDR に含まれる新しい VPC に割り当てられます。セキュリティグループは両方の新しい VPCs にクローンされますが、セキュリティグループルール CIDRs は自動的に更新されません。分割後にルールを確認して、VPC 間の通信が期待どおりに動作することを確認します。元の VPC は 2 つの新しい VPCs。

分割要件:

- 2 つの CIDR 範囲のみを指定する必要があります。
- 2 つの CIDRs 重複してはいけません。
- 各 CIDR は /16 から /28 の間である必要があります。
- すべてのサブネットは、2 つの CIDRs。サブネットが収まらない場合、オペレーションは拒否されます。

サブネットでは、次のオペレーションを使用できます。

- IP アドレスの変更: サブネット CIDR の基本 IP アドレスを変更しながら、プレフィックスの長さを同じにします。
- 削除: 親 VPC に影響を与えることなく、設定からサブネットを完全に削除します。
- サイズ変更: サブネット CIDR のプレフィックスの長さを変更して、IP アドレス範囲を拡大または縮小します。

サブネットのサイズ変更要件:

- 新しい CIDR は /16 から /28 の間である必要があります。
- 新しい CIDR は、同じ VPC 内の他のサブネットと重複してはいけません。
- 新しい CIDR は親 VPC CIDR 内にある必要があります。

各オペレーションの後、AWS Transform はセキュリティグループ参照を再評価します。これにより、CIDR ベースのルールをセキュリティグループ参照に変換したり、その逆を行ったりする可能性があります。変更を加えた後、セキュリティグループのルールを確認して、要件を満たしていることを確認します。

ガイド付きネットワークレコメンデーション

AWS Transform は、マッピングされたネットワークを分析し、チャットインターフェイスを通じて優先順位の高いレコメンデーションを提供します。推奨事項はネットワークデータに基づいており、変更を適用する前に確認する必要があります。

AWS 変換では、次の最適化が推奨される場合があります。

- CIDR 競合の解決: AWS 組織内のすべてのアカウントで、マッピング VPCs と既存の VPCs の間で CIDR 範囲が重複するフラグを立てます。競合 VPCs が最初に表示されます。マッピングされた VPC を再アドレスするか、除外または削除するか、競合を確認してデプロイ後に自分で解決することで、競合を解決できます。
- 命名標準化: 一貫したパターンに従わない VPC 名 (ハードウェアリファレンスを含む名前など) にフラグを付けます。AWS Transform は、置き換えを提案する前にクラウドの命名規則を要求します。
- スコープレビュー: レガシーシステムや廃止保留中のコンストラクトなど AWS、移行する必要のないネットワークセグメントを特定します。AWS Transform は、コンストラクトを除外する前に確認を求めます。
- VPC 容量の適切なサイズ設定: CIDR に含まれるサブネットのサイズが大きすぎるか小さすぎるように見える VPCs を表面化します。AWS Transform は現在の容量データを表示し、サイズを変更するかどうかを決定できます。
- セキュリティレビュー: レビューのために無制限のインバウンドトラフィック (0.0.0.0/0) を許可するセキュリティグループルールにフラグを付けます。
- VPC 統合: 論理的な分離要件ではなく、物理インフラストラクチャの制限で区切られたフラグメント化された VPCs を特定し、マージを提案します。

Note

すべてのレコメンデーションには、AWS Transform が変更を適用する前に明示的な確認が必要です。AWS Transform は、レコメンデーションがネットワークの複数の側面に影響を与える場合にトレードオフを提示します。

ステップ 6: ネットワーク図

生成された VPC 設定を確認したら、オプションでネットワーク図を生成してネットワークトポロジを視覚化できます。AWS Transform は次の図形式をサポートしています。

- Mermaid コード (.mmd): この形式は、Mermaid 互換ツールでレンダリングできるテキストベースの図定義ファイルを生成します。
- イメージ (.png): この形式は、ネットワークトポロジのレンダリングされたイメージを生成します。

ステップ 7: リソースのタグ付けを設定する

ネットワークリソースには、起動とレプリケーションのタグが付けられます。カスタムタグと AWS Migration Acceleration Program (MAP) タグを追加することもできます。

起動とレプリケーションの自動タグ

AWS 変換は、移行されたネットワークリソース (VPCs、サブネット、セキュリティグループ、ルートテーブル) に次のタグを自動的にタグ付けします。

- キー: CreatedBy 値: AWSApplicationMigrationService
- キー: ATWorkspace 値: *workspace-id*

これらのタグにより、VPC とサブネットを使用してテストインスタンスとカットオーバーインスタンスを起動できます AWS。

Note

移行された VPCs とサブネットにはデフォルトでインターネット接続が含まれていないため、レプリケーションのステージング領域として適していません。

また、VPC とサブネットをステージングエリア (レプリケーション) として使用するには、次のタグを手動で追加します。

- キー: CreatedFor 値: AWSTransform
- キー: ATWorkspace 値: *workspace-id*

これらのタグを既存の AWS ネットワークリソースに適用して、レプリケーションに使用できるようにすることもできます。

ウェブアプリの URL を AWS 変換する: [https://... /workspace/workspace-id/job/job-id](https://.../workspace/workspace-id/job/job-id)

カスタムタグ

AWS Transform によって自動的に適用されるタグに加えて、オプションでカスタムタグを追加して、移行されたネットワークリソースのコンプライアンスを整理、追跡、管理できます。カスタムタグは 2 つのレベルで適用できます。

- ジョブレベルのタグ: すべての VPCs、サブネット、セキュリティグループ、ルートテーブルなど、このジョブによって作成されたすべてのリソースに適用されます。
- VPC レベルのタグ: 特定の VPC に適用し、関連するすべてのリソース (サブネット、セキュリティグループ、ルートテーブル) に自動的にカスケードします。

Note

リクエストあたり最大 40 タグ。各タグにはキーと値が必要です。AWS タグ付け規則が適用されます。

AWS Transform は、Infrastructure as Code テンプレートを生成するときにこれらのタグを適用します。

AWS Migration Acceleration プログラム

移行が AWS Migration Acceleration Program (MAP 2.0) の一部である場合、AWS Transform はリソースに MAP タグを適用します。移行プロセスの前半で MPE ID を指定した場合、タグは自動的に適用されます。それ以外の場合、生成された VPC 設定の確認が完了すると、AWS Transform は MAP 契約があるかどうかを尋ね、大文字と数字 (ABCDE12345 など) を使用した 10 文字のコードである MPE ID の入力を求めます。適用されたタグは次の形式を使用します。

- キー: map-migrated 値: migMPE_ID

ステップ 8: ネットワークをデプロイする

タグ付け後、デプロイ戦略を選択します。

- AWS トランスフォームマネージドデプロイ: AWS トランスフォームは CloudFormation テンプレートを使用してネットワークをデプロイし、Reachability Analyzer を実行して、複数の VPCs 間および同じ VPC 内のサブネット間の接続をチェックします。

Note

ネットワークデプロイリクエストを実行する前に、明示的な承認を取得する必要があります。[「デプロイ承認プロセス」](#)を参照してください。

- 自己デプロイ: AWS Transform は、Infrastructure as Code (IaC) テンプレートを生成します。CloudFormation テンプレートはデフォルトで生成されます。追加の出力形式を選択することもできます。
- AWS CDK は、プログラムによるインフラストラクチャデプロイ用の TypeScript プロジェクトを生成します。
- HashiCorp Terraform は、ネットワークリソースを管理するための HashiCorp 設定言語 (HCL) テンプレートを生成します。
- Landing Zone Accelerator (LZA) は、LZA ネットワーク設定用の network-config.yaml ファイルを生成します。

Note

Landing Zone Accelerator (LZA) パイプラインを介してデプロイする場合、AWS Transform アカウントと LZA のインストールは同じ AWS Organization に存在する必要があります。Organization IDs。

自己デプロイの場合は、提供されたリンクを使用して、生成されたテンプレートを含む zip ファイルをダウンロードします。zip フォルダには、生成されたテンプレートの使用方法を説明する README.md ファイルが含まれています。

ダウンロードしたファイルが破損または改ざんされていないことを確認するには、チェックサムを生成してダウンロードし、`openssl dgst -sha256 -binary <file.zip> | base64` コマンドを使用してローカルに生成されたハッシュと比較します。

デプロイの承認プロセス

ネットワークの変更が組織のセキュリティ標準とアーキテクチャ要件に準拠していることを確認するために、すべてのデプロイリクエストは承認ワークフローを通過します。ネットワークデプロイリクエストを実行する前に、明示的な承認を取得する必要があります。デプロイリクエストを送信すると、AWS Transform Approvals タブを介して自動的に承認承認者にルーティングされます。承認者

は CloudFormation、テンプレートとネットワーク設定の両方を検証して、セキュリティ標準とアーキテクチャ要件への準拠を確認します。各送信は新しいレビューサイクルをトリガーし、デプロイは確認を受け取った後にのみ続行されます。承認者がリクエストを拒否した場合は、直接連絡して必要な変更について話し合います。AWS Transform は監査目的ですべての承認決定を追跡し、デプロイ履歴を維持します。

デプロイされたネットワークリソースを削除する

デプロイをロールバックする必要がある場合は、AWS Transform がデプロイしたネットワークリソースを削除できます。リソースは、デプロイが完了した直後に削除できます。デプロイ後にデプロイされたネットワークリソースを変更した場合、リソースを自動的に削除することはできません。

- AWS Transform-managed deployments: AWS Transform は、デプロイ中に作成されたすべての CloudFormation スタックを削除します。このアクションには、AWS Transform Approvals タブによる承認が必要です。
- 自己デプロイ: デプロイされたリソースは、AWS マネジメントコンソールまたは CLI AWS を使用して手動で削除する必要があります。

設定ファイルの抽出

ソース環境で Cisco ACL、Palo Alto Networks、または Fortinet FortiGate を使用している場合は、AWS Transform に提供する設定ファイルを抽出する必要があります。これらのファイルは、スタンドアロンのソースファイルとして使用してネットワークインフラストラクチャとセキュリティグループを生成したり、RVToolsファイルとして使用してセキュリティグループ生成を追加したりすることができます。抽出プロセスはどちらの場合も同じです。

ファイアウォール環境とネットワーク環境から設定ファイルを抽出するには、次の手順に従います。最新情報については、ベンダーのドキュメントを参照してください。

Fortinet FortiGate

- ファームウェアバージョンは v7.0 以降である必要があります。
- グローバルレベルで super_admin または super_admin_readonly 権限が必要です。
- 手順:
 1. SSH または組み込み CLI クライアント経由でファイアウォールに接続する
 2. 実行: `show | grep ""` (| grep "" ページ分割を無効にします)
 3. show コマンドからすべての出力をファイルに保存

Palo Alto Networks

- ファームウェアバージョンは 10.1 以降である必要があります。
- superadmin ロールが必要です。
- SSH 経由でファイアウォールに接続し、次のコマンドを実行してページ分割を無効にし、出力形式を設定し、設定モードに入り、設定と事前定義されたオブジェクトをエクスポートします。出力を保存します。

```
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

- ファームウェアバージョンは 6.0 以降である必要があります。
- すべての権限を持つ管理者ロールと、設定された Secure Copy Protocol (SCP)、SSH File Transfer Protocol (SFTP)、または File Transfer Protocol (FTP) の送信先が必要です。
- 手順:
 1. ブラウザから Application Policy Infrastructure Controller (APIC) に接続する
 2. 管理者メニューを開き、Config Rollbacks を選択します。
 3. 「スナップショットを作成する」ダイアログで、リモートロケーションオプションを選択し、「今すぐスナップショットを作成する」を選択します。
 4. 「転送成功」メッセージを受信したら、リモートロケーションサーバーに接続し、最新のスナップショットファイル (.gz ファイル) を取得します。

サーバーの移行

AWS Transform は AWS Transform MGN (MGN) を使用してサーバーを Amazon EC2 にリホストします。サーバー移行ワークフローは、各移行ウェブの設定、サーバーインベントリの検証、レプリケーションエージェントのデプロイ、データレプリケーションのモニタリング、移行されたインスタンスのテスト、最終的なカットオーバーの実行をガイドします。詳細については、[「とは」を参照してください AWS Transform MGN](#)。「MGN ユーザーガイド」の「」を参照してください。

サーバー移行はウェブ別に整理されています。各ウェブは、一緒に移行されるサーバーのグループを表します。ウェブごとに、次のフェーズを完了します。

コンテナ化移行戦略を使用するウェブの場合、AWS Transform は以下で説明するリホストステップの代わりにソースコードコンテナ化ワークフローを実行します。コンテナ化ワークフローでは、ソースコードのクローン作成、Docker アーティファクトの生成、コンテナイメージの公開、Amazon Elastic Container Service または Amazon Elastic Kubernetes Service へのデプロイについて説明します。完全なコンテナ化ワークフローについては、「」を参照してください[ソースコードのコンテナ化](#)。

1. 前提条件と移行デフォルトの設定
2. ステップ 1: 移行ウェブを設定する
3. ステップ 2: インベントリを検証して確認する
4. ステップ 3: レプリケーションエージェントをデプロイする
5. ステップ 4: データレプリケーション
6. ステップ 5: テスト
7. ステップ 6: カットオーバー

前提条件と移行デフォルトの設定

前提条件

リホスト移行を開始する前に、以下が整っていることを確認してください。

Note

AWS Transform でend-to-endの移行ジョブのすべてのステップを完了した場合、ターゲットアカウントとインベントリファイルは準備済みです。インベントリファイルは移行計画ステップ中に生成されます。AWS Transform ネットワーク移行を通じてセットアップされたネットワークインフラストラクチャも準備完了です。AWS Transform を使用してネットワークインフラストラクチャを構築しなかった場合は、リホスト移行を開始する前に事前に設定されていることを確認してください。

リホスト移行を開始する前に、サーバーをホストするためのネットワークリソースとインフラストラクチャが整っていることを確認します。AWS トランスフォームランディングゾーンとネットワーク移行機能、またはその他のツールを使用できます。

- サポートされているオペレーティングシステム – ソースサーバーは、サポートされているオペレーティングシステムを実行する必要があります。詳細なリストについては、「MGN ユーザーガイド」の「[サポートされているオペレーティングシステム](#)」を参照してください。
- 移行のターゲットアカウント – サーバーを移行する必要がある AWS アカウント IDs。トランス AWS フォームランディングゾーンまたはその他のツールを使用して、インフラストラクチャをセットアップできます。
- ネットワークインフラストラクチャの導入 – デプロイおよび設定された VPCs、サブネット、セキュリティグループ。AWS 変換ネットワーク移行やその他のツールを使用して、ネットワークインフラストラクチャをセットアップできます。
- インベントリファイル – サーバーの詳細、ウェーブ割り当て、ターゲットアカウント情報、Amazon EC2 インスタンスタイプの設定で準備されます。AWS 変換移行計画を使用して、このファイルを生成できます。

移行のデフォルトを設定する

マルチアカウント移行の実行を開始する前に、すべてのターゲットアカウントに適用されるデフォルト設定を設定する必要があります。これらのデフォルトは、Amazon EC2 インスタンスの起動方法と一般的な移行の設定方法を定義します。これらのデフォルトは、ウェーブのセットアップ中にウェーブレベルで上書きできます。

Amazon EC2 レコメンデーションの設定

AWS Transform は、ソース VMs の使用率仕様に基づいて Amazon EC2 インスタンスタイプのレコメンデーションを提供します。Amazon EC2 レコメンデーション設定を設定して、移行したサーバーでインスタンスタイプを選択する方法を制御できます。

Amazon EC2 レコメンデーションの生成の詳細については、「[」のAmazon EC2 レコメン AWS Migration Hub](#)デーションの生成」を参照してください。

Note

推奨される Amazon EC2 インスタンスタイプを変更して、[移行評価者](#)、[AWS 最適化とライセンス評価 \(OLA\)](#)、または AWS 変換評価ジョブからの推奨事項を含めることができます。

移行の初期化

移行を開始するには、AWS Transform は移行するすべての AWS リージョン と、サービスが使用されるすべてのターゲットアカウントに対して MGN を初期化します。初期化プロセス中:

- 必要な IAM ロールとポリシーが作成されます。
- 必要なデフォルトテンプレートが設定されています。

初期化プロセスの詳細については、「[MGN ユーザーガイド](#)」の「[コンソール AWS Transform MGN を使用した初期化](#)」を参照してください。

Amazon EC2 起動テンプレート

起動設定には、一般的な起動設定と Amazon EC2 起動テンプレートの 2 つの部分があり、ソースサーバーごとにテストインスタンスまたはカットオーバーインスタンスを起動する方法を決定します AWS。

Amazon EC2 起動テンプレートを含む起動設定は、アカウントレベルで定義でき、変換 MGN AWS にソースサーバーを追加するたびに自動的に各ソースサーバーに適用されます。このセクションで定義されている起動設定のデフォルトは、すべてのターゲットアカウントに自動的に適用できます。

AWS Transform は、使用可能な起動テンプレート設定のリストを表示します。デフォルトのまま続行するか、起動テンプレートを設定するかを選択できます。設定を選択した場合、AWS Transform は起動テンプレート設定のすべてのパラメータを含むヒューman-in-the-loop (HITL) レビューへのリンクを提供します。任意のパラメータについて、チャットインターフェイスから直接変更を加えることもできます。

[ソースサーバー](#)は、アカウント起動テンプレート設定で作成されます。これらのデフォルト設定でソースサーバーを作成したら、ソースサーバーの起動設定レベルで変更できます。チャットインターフェイスを使用して任意のパラメータのソースサーバー設定を変更するか、中にインベントリ Excel ファイルを使用して一括操作を行うことができます[ステップ 2: インベントリを検証して確認する](#)。

起動テンプレートの設定と詳細の完全なリストを確認するには、「[MGN ユーザーガイド](#)」の「[一般設定の起動](#)」を参照してください。

追加の Amazon EC2 起動テンプレートの変更

追加の Amazon EC2 起動テンプレートの変更については、各ターゲットアカウントのテンプレート ID で実行する必要があります。このオプションはウェーブ設定内で使用できます。AWS 変換は、ウェーブ設定をガイドし、適切なリンクを提供します。

ステップ 1: 移行ウェーブを設定する

このフェーズでは、AWS Transform はターゲットアカウントの設定、サービスアクセス許可の検証、リソースタグの設定、インベントリへのネットワークデータの追加、レプリケーションと起動の設定を行うことで、移行ウェーブを準備します。

移行モードとアカウント設定

AWS Transform は 2 つの移行モードをサポートしています。

- シングルアカウント移行 – ウェーブ内のすべてのサーバーは、コネクタで設定された同じターゲットアカウントに移行します。
- マルチアカウント移行 – サーバーは、インベントリファイルで指定された異なるターゲットアカウントに移行します。マルチアカウント移行の場合、インベントリファイルには、各サーバーのターゲットアカウント ID を含む `mgn:account-id` 列が含まれている必要があります。

AWS Transform はターゲットアカウント設定を確認し、MGN が各ターゲットアカウントで初期化されていることを確認します。MGN がまだ初期化されていない場合、AWS Transform は初期化を完了する手順を提供します。初期化中、MGN はレプリケーションおよび起動オペレーション用に次の IAM サービスロールを作成します。

- `AWSApplicationMigrationReplicationServerRole`
- `AWSApplicationMigrationConversionServerRole`
- `AWSApplicationMigrationMGHRole`
- `AWSApplicationMigrationLaunchInstanceWithDrsRole`
- `AWSApplicationMigrationLaunchInstanceWithSsmRole`
- `AWSApplicationMigrationAgentRole`

これらのロールの詳細については、[「MGN ユーザーガイド」の「コンソールを使用した MGN の初期化」](#)または[「API を使用した MGN の初期化」](#)を参照してください。

マルチアカウント移行の場合、AWS Transform は初期化ステップ中に次のロールも作成します: `AWSTransformRehostSharingRole_<management-or-delegated-admin-account-id>`。このロールは、すべての移行ターゲットアカウントにデプロイされます。

リソースのタグ付けの検証

サービスアクセス許可が確認されると、AWS Transform は、エージェントが正常に移行を運用するために必要なすべてのリソースに適切にタグ付けされていることを確認します。必要なタグが欠落しているリソースがある場合、AWS 変換はタグ付けページへのリンクを提供します。ここでは、続行する前に欠落しているタグを適用できます。次のタグが必要です。

- 既存のソースサーバーにはタグ `CreatedBy: AWSTransform` と `必要ですATWorkspace: <workspace_id>`。ソースサーバーでレプリケーションをすでに開始し、AWS Transform MGN サービスで作成している場合は、AWS Transform がオンプレミス環境から検出されたソースサーバーと関連付けられるように、これらのサーバーにタグを付ける必要があります。これにより、ソースサーバーが重複するのを防ぐことができます。AWS Transform は、ユーザーが指定した ID、FQDN、またはホスト名キーを使用して、それらのサーバーを自動的に関連付けます。
- ネットワークリソースは、レプリケーション (ステージングエリア) と起動インスタンスの両方で適切にタグ付けする必要があります。AWS Transform は、ターゲットアカウントのネットワークリソースの完全なリストを表示し、各リソースに既にタグ付けされているかどうかを示します。リストを確認し、追加するタグのないリソースを選択できます。選択したリソースごとに、AWS Transform は関連するタグを適用します。
- `CreatedBy: AWSTransform` または `。リソースタイプCreatedFor: AWSTransform` によって異なります。
- `ATWorkspace: <workspace_id>` は、選択したすべてのリソースに適用されます。

変換ネットワーク移行エージェントによって作成された VPCs AWS とサブネットには、自動的にタグが付けられます。

- VPCs とサブネットに加えて、AWS Transform にはターゲットアカウントで見つかった既存のすべての Elastic Network Interface (ENIs) も表示されます。AWS Transform でインスタンスの起動の一部として使用する場合は、`CreatedFor: AWSTransform` と `でタグ付けする必要がありますATWorkspace: <workspace_id>`。Amazon EC2 起動テンプレートに ENIs [「詳細な考慮事項」](#) を参照してください。

ネットワークデータをインベントリに追加する

AWS Transform は、ネットワーク移行からのネットワーク情報をインベントリファイルに追加します。このステップでは、移行ネットワークフェーズ中に生成されたネットワーク設定に基づいて、サーバーを適切なターゲットサブネットとセキュリティグループにマッピングします。

レプリケーションと起動の設定

レプリケーション設定の構成

レプリケーション設定は、ソースサーバーからデータをレプリケートする方法を決定します AWS。Transform MGN AWS にソースサーバーを追加する前に、レプリケーションテンプレートでレプリケーション設定を構成します。AWS 変換では、すべてのレプリケーション設定パラメータが表示されます。専用の HITL またはチャットインターフェイスを使用して設定できます。

レプリケーション設定パラメータの詳細については、「MGN ユーザーガイド」の [「レプリケーション設定テンプレート」](#) を参照してください。

起動テンプレートの設定

起動テンプレートを使用すると、AWS Transform MGN がインスタンスを起動する方法を制御できます AWS。テンプレートで定義されたデフォルト設定は、新しく追加されたすべてのサーバーに自動的に適用されます。起動テンプレートの設定は、専用の HITL またはチャットインターフェイスを使用して設定できます。

起動テンプレート設定パラメータの詳細については、MGN ユーザーガイドの [「起動テンプレート」](#) を参照してください。

AWS Transform は、起動テンプレートに関連付けられた Amazon EC2 起動テンプレート ID へのリンクも提供するため、追加の Amazon EC2 起動テンプレート属性を変更できます。Amazon EC2 起動テンプレートを編集するには、「MGN ユーザーガイド」の [「起動テンプレート」](#) の指示に従います。

IP 割り当て戦略

移行したサーバーに IP アドレスを割り当てる方法を選択します。

- 静的 IP – ソースサーバーの IP アドレスは維持されます。CIDR 変換が必要な場合、AWS 変換は新しい CIDR と一致するように IP アドレスを自動的に変換します。
- 動的 IP (DHCP) – 各サーバーには、サブネットの IP プールから新しい IP アドレスが割り当てられます。

Note

ネットワーク移行中に MAP セキュリティグループマッピング戦略を選択した場合は、静的 IP 割り当てのみを使用できます。詳細については、[「セキュリティグループのマッピング」](#)を参照してください。

ステップ 2: インベントリを検証して確認する

サーバーデータを MGN にロードする前に、AWS Transform はインベントリファイルをレビュー用に準備します。CSV または XLSX 形式でファイルをダウンロードし、サーバー設定を確認し、必要に応じて変更を加えることができます。

インベントリファイルには、サーバー名、オペレーティングシステム、Amazon EC2 インスタンスタイプの推奨事項、ターゲットサブネット、セキュリティグループ、IP 割り当て、ライセンスオプションなどの詳細が含まれます。必須フィールドは次のとおりです。

- サーバー情報 – サーバー名、VMID、およびソース仕様。
- ウェーブ割り当て — 移行ウェーブのグループ化。
- アプリケーショングループ化 – 論理アプリケーションの関連付け。
- ターゲット設定 – ターゲットアカウント、リージョン、Amazon EC2 インスタンスタイプ。
- ネットワーク設定 — ターゲットサブネットとセキュリティグループ。

ファイルを変更して、Amazon EC2 設定の調整、オペレーティングシステムのライセンスオプション (BYOL またはライセンス込み) の変更、テナンシー設定の更新を行うことができます。

インベントリを確認したら、次のように受け入れるか、変更されたバージョンをアップロードできます。AWS Transform はデータを MGN にロードし、ウェーブ内の各サーバーのソースサーバーレコードを作成します。

Note

インベントリファイルの列を削除したり、列ヘッダーを変更したりしないでください。AWS 変換では、データを正しく処理するために元のファイル構造が必要です。

Note

AWS 変換では、特定のターゲット AWS アカウント とターゲットに AWS リージョン 一度に 1 回インポートできます。複数のウェーブで同時に作業する場合、または同じターゲットアカウントで複数の移行ジョブが実行されている場合は、インポートが終了するのを待ってから、別のウェーブまたはジョブで別のインポートを実行する必要があります。

インベントリファイルの列と で設定を指定することで、オペレーティングシステムのライセンスオプション (BYOL またはライセンス込み) `mgn:launch:placement:operating-system-licensing`とテナンシーを制御できます`mgn:launch:placement:tenancy`。詳細については、「MGN ユーザーガイド」の「[パラメータのインポート](#)」を参照してください。

ステップ 3: レプリケーションエージェントをデプロイする

ソースサーバーから へのデータのレプリケーションを開始するには AWS、各ソースサーバーに AWS レプリケーションエージェントをインストールします。AWS Transform には 3 つのインストール方法があります。

- 組織ツール – 組織の既存のデプロイツール (SCCM、Ansible、Chef など) を使用して、サーバー全体にエージェントをインストールします。AWS Transform は、`--no-prompt`、`--aws-access-key-id`、`--aws-secret-access-key`などのサイレントインストール用の追加のパラメータを含むインストールコマンドを提供します`--aws-session-token`。
- MGN コネクタ – MGN コネクタを使用してエージェントのインストールを自動化します。コネクタは、SSH (Linux) または WinRM (Windows) を介してソースマシンに接続し、レプリケーションエージェントを自動的にインストールします。設定すると、コネクタを複数のウェーブと異なるターゲットで再利用できます AWS アカウント。MGN コネクタの詳細については、「[MGN ユーザーガイド](#)」の「[MGN コネクタのセットアップ](#)」を参照してください。

Note

AWS Transform で MGN コネクタを使用する前に、Fleet Manager AWS Systems Manager のコネクタのマネージドインスタンスに次のタグを付ける必要があります。

- キー: `CreatedFor` 値: `AWSTransform`
 - キー: `ATWorkspace` 値: `workspace-id`
- マネージドインスタンスにタグを付けるには、AWS Systems Manager コンソールを開き、Node Tools で Fleet Manager に移動し、MGN コネクタのマネージドインスタンスを

選択し、上記のタグを適用します。ウェブアプリ URL の AWS 変換: でワークスペース ID を見つけます `https://.../workspace/workspace-id/job/job-id`。

- 手動インストール – エージェントを各ソースサーバーに直接インストールします。この方法では、各サーバーに直接アクセスする必要がありますが、インストールプロセスを完全に制御できます。

AWS MGN コネクタのセットアップを変換する

AWS Transform MGN コネクタは、レプリケーションエージェントのソースサーバーへのデプロイを自動化します。コネクタは、オンプレミス環境の専用 Linux マシンにデプロイされた軽量クライアントです。SSH (Linux) または WinRM (Windows) 経由でソースサーバーに接続してレプリケーションエージェントをインストールおよび設定するため、複数の AWS サービス間で手動で調整する必要がなくなります。

コネクタの仕組み

コネクタは、次のコンポーネントを介して動作します。

- コネクタクライアント – 環境内の専用 Linux マシンにデプロイされます。
- SSM エージェント – 同じマシンにインストールされ、との安全な通信を可能にします AWS。
- SSM Hybrid Activation – コネクタマシンを AWS Systems Manager にリンクして、安全なコマンドを実行します。
- 認証情報管理 – AWS Secrets Manager からソースサーバーの認証情報を取得します。

エージェントをデプロイすると、AWS Transform はコネクタマシンに SSM ドキュメントを送信します。次に、コネクタは AWS Secrets Manager からソースサーバーの認証情報を取得し、各ソースサーバーへの接続を確立し、ソースサーバーが前提条件を満たしていることを検証し、レプリケーションエージェントをインストールして設定し、正常にインストールされたことを確認します。

コネクタマシンの要件

要件	詳細
オペレーティングシステム	サポートされている Linux オペレーティングシステム。詳細なリストについては、「 MGN ユーザーガイド 」の「 MGN コネクタの前提条件 」を参照してください。

要件	詳細
ネットワークアクセス	すべてのソースサーバーに到達する必要があります (SSH 経由の Linux、WinRM 経由の Windows)
インターネット接続	AWS エンドポイントへのアウトバウンド HTTPS (443) (Systems Manager、Secrets Manager、MGN)
ディスク容量	最小 200 MB 無料
アクセス許可	ルートまたは sudo アクセス

Note

コネクタは Linux マシンにインストールする必要がありますが、Linux と Windows の両方のソースサーバーにエージェントをデプロイできます。

セットアッププロセス

AWS Transform は、コネクタをセットアップするための次のステップをガイドします。

ステップ 1: コネクタ設定

コネクタの名前を指定するか、自動生成されたデフォルト名を使用します。コネクタは、管理アカウントまたは MGN の委任管理者アカウントにインストールできます。マルチアカウント移行の場合、コネクタはメンバーアカウント間でエージェントをサーバーにデプロイできます。

ステップ 2: AWS リソースのセットアップ

AWS Transform は、AWS 認証情報を使用してブラウザで実行されるセットアップページを開きます。AWS 管理アカウントまたは委任された管理者アカウントを使用して、マネジメントコンソールにログインする必要があります。これは、AWS 変換ターゲットコネクタが接続されているアカウントと同じである必要があります。

セットアップページでは、次のリソースが自動的に作成されます。

- IAM ロール (幕等の作成 — 既に存在する場合はスキップ):
 - `AWSApplicationMigrationConnectorManagementRole` – 認証情報にアクセスするためにエージェントのインストール中に使用されます。

- `AWSApplicationMigrationConnectorSharingRole_<ACCOUNT-ID>` – エージェントのインストールのアクセス許可が含まれます。
- SSM Hybrid Activation – 30 日間の有効期間。コネクタマシン AWS を Systems Manager にリンクし、安全なアクティベーション認証情報を生成します。

または、セットアップページから CloudFormation テンプレートをダウンロードして、IAM ロールを自分でデプロイすることもできます。

セットアップページは、必要なすべての認証情報と設定を含む 1 行のインストールコマンドを生成します。

Important

インストールが完了するまで、セットアップページを開いたままにします。これを閉じるには、プロセスを再起動する必要があります。すべての認証情報はブラウザにのみ存在し、AWS 変換によって保存されません。

ステップ 3: コネクタのインストール

環境の Linux マシンにコネクタをインストールします。

1. セットアップページからインストールリンクをコピーします。
2. 選択した Linux マシンに SSH します。
3. インストールコマンドを貼り付けて実行します。
4. インストールが完了するまで待ちます (通常は 2~3 分)。

ステップ 4: ソースサーバーをアタッチする

インストール後、AWS Transform は現在のウェブに属するすべてのソースサーバーを識別し、自動的に MGN コネクタにアタッチします。

ステップ 5: 認証情報を設定する

ソースサーバーの認証情報に AWS Secrets Manager ARNs を指定します。AWS Transform には 3 つの認証情報設定オプションがあります。

- Linux サーバーの単一のシークレット – すべての Linux ソースサーバーの SSH キーまたはユーザー名/パスワードを含む 1 つの共有シークレット。
- Windows サーバー用の 1 つのシークレット – すべての Windows ソースサーバーのユーザー名とパスワードを含む 1 つの共有シークレット。
- サーバーごとの複数のシークレット – サーバーまたはサーバーのグループごとに異なるシークレット。サーバーに異なる認証情報がある場合に使用します。AWS Transform は、サーバーリストがあらかじめ入力された CSV ファイルを生成します。各サーバーの `secret_arn` 列に入力し、完成したファイルをアップロードします。

Note

両方のサーバータイプにそれぞれ 1 つの共有シークレットがある場合、Linux と Windows の単一シークレットオプションを組み合わせることができます。サーバーごとのシークレットオプションは、単一シークレットオプションと相互に排他的です。

認証情報シークレット形式。詳細については、[「MGN ユーザーガイド」の「MGN コネクタの認証情報」](#)を参照してください。

```
{
  "WinConnectionProtocol": "HTTPS",
  "WinUserName": "windows_username",
  "WinPassword": "windows_password",
  "LinuxUserName": "linux_username",
  "LinuxPrivateKey": "linux_private_key",
  "LinuxHostKeyValidation": false
}
```

エージェントデプロイ

認証情報が設定および検証されると、AWS Transform はレプリケーションエージェントをソースサーバーにデプロイします。現在のウェーブ内のすべてのサーバーにデプロイすることも、特定のサーバーを選択することもできます。

各サーバーのデプロイプロセス:

1. AWS Transform は、SSM を介してコネクタにデプロイコマンドを送信します。
2. コネクタは AWS Secrets Manager から認証情報を取得します。

3. コネクタは、設定された認証情報を使用してソースサーバーに接続します。
4. コネクタは、ソースサーバーがレプリケーションエージェントの実行に必要なすべての前提条件を満たしていることを確認します。
5. コネクタはレプリケーションエージェントをインストールして設定します。
6. コネクタは、正常なインストールと接続を検証します。

現在のインストールステップ、経過時間、推定残り時間など、サーバーごとのステータス追跡を使用して、デプロイの進行状況をリアルタイムでモニタリングできます。サーバーに障害が発生した場合、AWS Transform は障害の理由を表示し、サーバーごとに再試行オプションを提供します。正常にデプロイされたサーバーは、障害が発生したサーバーが再試行されている間、個別に続行できます。

コネクタの再利用とライフサイクル

後続のウェーブ用にエージェントをデプロイするときは、既存のコネクタを再利用するか、新しいコネクタを作成できます。AWS Transform は、アカウントで設定されたすべてのコネクタを一覧表示し、コネクタ名、ステータス (アクティブまたは期限切れ)、アタッチされたサーバー数、ハイブリッドアクティベーションの有効期限を示します。

- アクティブなコネクタ – Hybrid Activation は引き続き有効です。AWS Transform は新しいウェーブの IAM ロールを検証し、認証情報設定に進みます。新しいハイブリッドアクティベーションは必要ありません。
- コネクタの有効期限切れ – SSM Hybrid Activation の有効期限が切れています。期限切れのアクティベーションは更新できません。別のコネクタを選択するか、新しいコネクタを作成する必要があります。

SSM Hybrid Activations は 30 日後に期限切れになります。アクティベーションは、Linux マシンにコネクタをインストールする場合にのみ必要です。コネクタがインストールされたら、アクティベーションの有効期限が切れた後でも、引き続きそれを使用してレプリケーションエージェントをソースサーバーにインストールできます。アクティベーションの有効期限が切れた後にコネクタを新しいマシンにインストールする必要がある場合は、セットアッププロセスを通じて新しいコネクタを作成する必要があります。

エージェントの手動インストール

手動インストールでは、まず AWS 認証情報 (一時的または永続的) を生成し、次に各ソースサーバーにエージェントをインストールします。

認証情報オプション:

- 一時的な認証情報 (推奨) – AWSApplicationMigrationAgentInstallationPolicy 管理ポリシーを使用して IAM ロールを作成し、を使用して一時的な認証情報 `aws sts assume-role` を生成します。詳細については、「MGN ユーザーガイド」の [「エージェントのインストール許可」](#) を参照してください。
- 永続的認証情報 – AWSApplicationMigrationAgentInstallationPolicy 管理ポリシーを使用して IAM ユーザーを作成し、アクセスキーを生成します。

インストール手順:

Linux サーバーの場合は、インストーラをダウンロードして実行します。

```
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
```

Windows サーバーの場合は、PowerShell を管理者として使用して、適切なインストーラをダウンロードして実行します。

```
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

--user-provided-id パラメータは必須です。*server-identifier* をインベントリファイルの `mgn:server:user-provided-id` 列の正確な値に置き換えます。この識別子は、物理サーバーを MGN ソースサーバーレコードにリンクします。

エージェントのインストールの詳細については、「MGN ユーザーガイド」の [「Linux エージェント」と \[Windows エージェント\]\(#\)](#)」を参照してください。

インストール後、AWS Transform は、サーバーに INITIATING または のレプリケーション状態が表示されていることを確認して、すべてのエージェントが正常に接続されていることを確認します INITIAL_SYNC。

Note

AWS 変換は MGN エージェントレスレプリケーションをサポートしていません。エージェントレスレプリケーションの詳細については、「MGN ユーザーガイド」の「[エージェントレスレプリケーションの概要](#)」を参照してください。

Note

ウェブ内のすべてのサーバーにレプリケーションエージェントをインストールする必要があります。レプリケーションエージェントをインストールしないサーバーを切断してアーカイブします。disconnect-from-service コマンドを使用してサーバーを切断し、mark-as-archived コマンドを使用して切断されたサーバーをアーカイブできます。アーカイブコマンドは、ライフサイクル状態が であるソースサーバーでのみ機能します DISCONNECTED。

レプリケーションに関連するクォータについては、「[MGN ユーザーガイド](#)」の「[MGN サービスクォータの制限](#)」を参照してください。

ステップ 4: データレプリケーション

レプリケーションエージェントをインストールすると、データレプリケーションが自動的に開始されます。AWS Transform は継続的なブロックレベルのレプリケーションを使用して、ソースサーバーから にデータを同期します AWS。

レプリケーションプロセスは 2 つのフェーズで構成されます。

- 初期同期 — ソースサーバーデータの完全コピー AWS。データは、設定されたターゲットストレージタイプに応じて、Amazon Elastic Block Store (Amazon EBS) スナップショットとして、またはターゲットアカウントの Amazon FSx for NetApp ONTAP (FSx for ONTAP) ボリュームに保存されます。詳細については、「MGN ユーザーガイド」の「[ターゲットストレージタイプ](#)」を参照してください。期間は、データ量とネットワーク帯域幅によって異なります。

- 継続的レプリケーション – ソースサーバーのパフォーマンスへの影響を最小限に抑えながら、変更されたブロックの継続的な同期を行います。up-to-dateコピーを維持します AWS。

レプリケーションサーバーは、ステージングエリアサブネットにデプロイされた一時的な Amazon EC2 インスタンスです。ソースサーバーからレプリケートされたデータを受信し、MGN によって自動的に管理されます。詳細については、MGN ユーザーガイドの「[レプリケーションサーバーの設定](#)」を参照してください。

AWS Transform はレプリケーションの進行状況をモニタリングし、レプリケーションのステータス、レプリケーションの遅延 (ソースデータとレプリケートされたデータの時間差)、帯域幅の使用状況など、ステータスの更新を提供します。

レプリケーション中、各サーバーは次の状態に進みます。

- 準備中 – サーバーは初期同期プロセス中であり、まだテストする準備ができていません。
- テスト準備完了 – サーバーが正常に追加され、データレプリケーションが開始されました。テストインスタンスまたはカットオーバーインスタンスを起動できるようになりました。

ウェーブ内のすべてのサーバーが NOT_READY状態を超えると、データレプリケーションフェーズが完了し、テストに進むことができます。

個々のサーバーまたはウェーブ全体のレプリケーションはいつでも制御できます。

- レプリケーションの一時停止 – 特定のサーバーまたはウェーブ全体のレプリケーションを一時的に一時停止します。
- レプリケーションを再開する – 以前に一時停止したレプリケーションを再開します。
- レプリケーションを停止する – レプリケーションを完全に停止します。停止したレプリケーションは再起動できますが、最初の同期から開始されます。

ステップ 5: テスト

データレプリケーションが完了したら、テストインスタンスを起動して、移行したサーバーを検証してから、最終的なカットオーバーを実行できます。詳細については、「MGN ユーザーガイド」の「[テストインスタンスの起動](#)」を参照してください。AWS Transform は 2 つのテストオプションをサポートしています。

- フルウェーブテスト – ウェーブ内のすべてのサーバーに対してテストインスタンスを起動します。
- 選択的テスト – インベントリファイルからユーザー提供IDs を指定して、選択した特定のサーバーのテストインスタンスを起動します。

AWS Transform は、レプリケートされたデータから Amazon EC2 インスタンスを起動し、テストインスタンスに接続して検証できるようにインスタンス IDs を提供します。テスト後、次のことができます。

- テストが成功したらカットオーバーに進みます。
- 新しいテストインスタンスを起動して再テストします。
- テストインスタンスを終了し、再テストする前に問題に対処します。

ステップ 5b: アプリケーションをカットオーバー準備完了としてマークする

テストが完了し、結果に満足したら、アプリケーションをカットオーバーの準備ができているものとしてマークします。AWS Transform は各アプリケーションのレプリケーションステータスを確認し、レプリケーションアラートを解決してから続行します。カットオーバーとしてマークできるのは、クリーンレプリケーションステータスのアプリケーションのみです。

ステップ 6: カットオーバー

カットオーバーは、本番ワークロードが移行される最後の移行ステップです AWS。詳細については、MGN ユーザーガイドの [「カットオーバーインスタンスの起動」](#) を参照してください。テストと同様に、AWS Transform は特定のサーバーに対してフルウェーブカットオーバーまたは選択的カットオーバーをサポートしています。

カットオーバー中、AWS Transform は最新のレプリケートされたデータから Amazon EC2 インスタンスを起動し、各サーバーのインスタンス IDs を提供します。カットオーバーインスタンスを確認したら、カットオーバーを確定し、進行中のソースマシンのレプリケーションを停止します。

カットオーバープロセスには、以下のステップが含まれます。

1. カットオーバーインスタンスの起動 – AWS 選択したサーバーの Amazon EC2 インスタンスを起動します。フルウェーブカットオーバーまたは選択的カットオーバーを選択できます。

2. カットオーバーインスタンスの検証 – 起動したインスタンスに接続し、正しく機能していることを確認します。
3. カットオーバーを確定する – カットオーバーを確認してソースマシンのレプリケーションを停止します。ウェーブ内のすべてのサーバーを確定するか、特定のサーバーを選択できます。ファイナライゼーションは、レプリケーションエージェントがデータを送信するのを停止し、レプリケーションエージェントをソースサーバーから削除して、サーバーのライフサイクル状態をロックします。このアクションは簡単に元に戻すことはできません。詳細については、「MGN ユーザーガイド」の「[カットオーバーの完了](#)」を参照してください。
4. アーカイブソースサーバー (オプション) – 確定後、ソースサーバーをアーカイブ済みとしてマークして、アカウントのソースサーバーのクォータを解放できます。

Important

カットオーバーを確定すると、進行中のソースマシンのレプリケーションが停止します。確定する前に、カットオーバーインスタンスを検証していることを確認してください。

Note

ダウンタイムは、ソースのシャットダウンとカットオーバーインスタンスの可用性の間で発生します。それに応じてカットオーバーウィンドウを計画します。

サーバーライフサイクルの状態

移行中、各サーバーは次のライフサイクル状態に進みます。詳細については、MGN ユーザーガイドの「[ソースサーバーのライフサイクル](#)」を参照してください。

- 準備中 – サーバーは初期同期プロセス中であり、まだテストする準備ができていません。
- テスト準備完了 – データレプリケーションが開始され、テストインスタンスまたはカットオーバーインスタンスを起動できます。
- テスト中 – テストインスタンスは現在起動中です。
- カットオーバーの準備 – サーバーはテスト済みで、カットオーバーの準備ができました。
- カットオーバー中 – カットオーバーインスタンスは現在起動中です。

- カットオーバー完了 – サーバーはカットオーバーされました。すべてのデータが AWS カットオーバーインスタンスに移行されました。
- Disconnected – サーバーが MGN から切断されました。

移行中はいつでも、サーバーのステータスについて AWS Transform に尋ねることができます。AWS Transform には、移行ライフサイクル、レプリケーションステータス、推奨される次のステップなど、関連するすべてのサーバー情報を表示するインタラクティブなウェブステータステーブルが用意されています。自然言語で質問することもできます。次に例を示します。

- サーバーのステータスを教えてください。
- ウェーブのステータスを教えてください。
- 現在使用しているステップのステータスを教えてください。

ウェーブ移行中に、AWS Transform に個々のサーバーのステータスを更新または変更するように依頼できます。例えば、ウェーブ内の 10 台のサーバーのうち 9 台がテストフェーズに合格したが、1 台が失敗した場合、AWS Transform は失敗したサーバーでテストを再実行しながら、9 台のサーバーを次のフェーズに移動し続けることができます。

デプロイの承認

一部の移行オペレーションでは、実行前に明示的な承認が必要です。オペレーションで承認が必要な場合、AWS Transform は承認タブを通じてリクエストを承認された承認者にルーティングします。AWS Transform の管理者ロールを持つユーザーのみがデプロイリクエストを承認できます。デプロイは、確認を受け取った後にのみ続行されます。

リリースノート

次のリリースノートでは、に対する最新の変更について説明します[移行 \(VMware を含む\)](#)。サポートされている AWS 変換リージョンについては、[「サポートされているリージョン」](#)を参照してください。サポートされているターゲットリージョンについては、[アカウントコネクタの設定ページ](#)を参照してください。

2026 年 6 月

- AWS 移行の変換でローカリゼーションがサポートされるようになりました。移行ワークフローを操作するときに、ウェブアプリケーションの表示言語を変更できます。[言語設定の詳細について説明します](#)。

- AWS Transform では、移行用にレプリケーション設定と起動設定を設定できるようになりました。ターゲットアカウントごと、または特定のソースサーバー間で設定を設定できます。[レプリケーションと起動の設定について説明します](#)。
- AWS Transform では、既存の Elastic Network Interface (ENIs) を起動テンプレートにアタッチできるようになりました。ターゲットアカウントの ENIs にタグを付けると、インスタンスの起動時に使用できるようになります。[ネットワークリソースのタグ付けについて説明します](#)。
- AWS 移行の変換は、中東 (バーレーン) と中東 (アラブ首長国連邦) を除くすべての AWS 商用リージョンを移行ターゲットとしてサポートするようになりました。[サポートされているターゲットリージョンの詳細について説明します](#)。

2026 年 5 月

- AWS Transform VPCs を検出するようになりました。マッピングVPCs と一緒に既存の VPCs を表示し、CIDR 競合を特定して、デプロイ前に解決できます。[既存の VPC 検出の詳細について説明します](#)。
- AWS Transform は、移行中のソースコードリポジトリのコンテナへのリプラットフォームをサポートしています AWS。[コンテナ化の詳細について説明します](#)。

2026 年 4 月

- 移行ワークフロー内でランディングゾーンの作成を直接追加しました。既存の基盤のチェックを AWS 変換し、組織単位 (OU) 構造とターゲットアカウントを推奨し、自動デプロイまたは Infrastructure as Code (IaC) 出力 (CloudFormation AWS CDK、または Landing Zone Accelerator (LZA) 形式) を提供します。[ランディングゾーンの作成について詳しく説明します](#)。
- ネットワーク移行中のセキュリティグループマッピングによる DHCP のサポートを追加しました。[セキュリティグループの作成について詳しく説明します](#)。

2026 年 3 月

- ネットワーク移行APIs が追加され、パートナーとプログラムによるネットワーク移行機能へのアクセスが可能になりました。[ネットワーク移行 API リファレンス](#)を参照してください。

2026 年 2 月

- ネットワーク移行では、[RVTools 検出データを使用せずに、ファイアウォールまたは Software-Defined Networking \(SDN\) 設定ファイルの](#)取り込みがサポートされるようになりました。ファイアウォールまたは SDN 設定ファイルを個別に使用することも、RVTools エクスポートと組み合わせることもできます。

2026 年 1 月

- [移行促進プログラム \(MAP\) のタグ付けとネットワーク移行要素でのユーザー定義のタグ付けの](#)サポートが追加され、MAP クレジットと組織目的で移行リソースを追跡できるようになりました。

2025 年 12 月

- ワークフローの再設計と使いやすさの向上により、新しい[移行エクスペリエンス](#)が追加されました。詳細については、「[Accelerating VMware migration: AWS Transform's new experience](#)」を参照してください。
- 自動ランディングゾーン設定用のランディングゾーンアクセラレーター (LZA) ネットワーク設定を追加しました。
- 移行実行時の[複数のターゲット AWS アカウント](#)のサポートが追加され、アカウント間でワークロードを移行できるようになりました。
- [Cisco ACI、Palo Alto、FortiGate、ModelizeIT ソース形式の](#)ネットワーク移行サポートを追加しました。

2025 年 10 月

- アジアパシフィック (大阪) を追加し、移行実行を [16 のターゲットリージョン](#)に拡張しました。

2025 年 9 月

- [リホストウェーブ](#)でのライセンス設定のサポートが追加され、ウェーブごとに Bring Your Own License (BYOL) またはライセンス込みオプションを指定できるようになりました。
- [ネットワーク移行で Terraform](#) のサポートが追加され、Terraform テンプレートを使用してネットワークインフラストラクチャをデプロイできるようになりました。

- レプリケーションサブネット設定の更新のサポートが追加され、ターゲットアカウントのデフォルト VPC の要件が削除されました。

2025 年 8 月

- 米国東部 (オハイオ)、欧州 (ストックホルム)、欧州 (アイルランド) を追加し、移行の実行 (ネットワーク移行とリホスト) を [15 のターゲットリージョン](#) に拡張しました。
- ネットワーク移行用の [柔軟な CIDR 範囲機能](#) が追加され、ネットワークのデプロイ中に IP アドレス範囲をカスタマイズできるようになりました。

2025 年 5 月

- の初期サービス起動。 [検出移行 \(VMware を含む\)](#)、評価、移行計画、ネットワーク移行、サーバーリホストなどの end-to-end の移行機能を提供します。 <https://docs.aws.amazon.com/transform/latest/userguide/transform-vmware-review-groupings-and-waves.html>
- 米国東部 (バージニア北部)、米国西部 (オレゴン)、カナダ (中部)、南米 (サンパウロ)、欧州 (フランクフルト)、欧州 (ロンドン)、欧州 (パリ)、アジアパシフィック (ムンバイ)、アジアパシフィック (ソウル)、アジアパシフィック (東京)、アジアパシフィック (シンガポール)、アジアパシフィック (シドニー) の 12 の [ターゲットリージョン](#) でサポートされている移行の実行。

ソースコードのコンテナ化

AWS Transform は、移行中のアプリケーションのコンテナへのリプラットフォームをサポートしています。AWS。この章では、AWS トランスフォームのエージェント AI 機能について説明し、ソースコードのコンテナ化を自動化します。並行して移行およびモダナイズできるため、オンプレミスからクラウドネイティブアーキテクチャへの移行にかかる時間と複雑さが軽減されます。GitHub、Bitbucket、GitLab、または .zip ファイルからソースコードをコンテナ化したり、Docker イメージを生成したり、Amazon Elastic Container Registry (Amazon ECR) に発行したり、Amazon Elastic Container Service (Amazon ECS) または Amazon Elastic Kubernetes Service (Amazon EKS) にデプロイしたりできます。これにより、コンテナ化は、リホスト移行の計画と実行に使用すると同じワークフローになります。

Note

ソースコードのコンテナ化は、まだコンテナ化されていないアプリケーション向けに設計されています。アプリケーションのソースコードにアクセスする必要があり、既存のコンテナ化されたワークロードの移行はサポートされていません。既存のコンテナを に移行するには AWS、Amazon Elastic Container Service または Amazon Elastic Kubernetes Service の標準デプロイ方法を使用します。

機能と主要機能

AWS Transform には、アプリケーションをコンテナ化するための以下の機能があります。

- AI 駆動型ソースコード分析 — アプリケーションのソースコードを分析し、Dockerfile や関連する設定ファイルなどの Docker アーティファクトを自動的に生成します。
- コンテナイメージの構築と公開 — AI 主導の Docker ビルドを使用してコンテナイメージを構築およびテストし、自動脆弱性スキャンを使用して Amazon Elastic Container Registry に公開します。
- コードとしてのインフラストラクチャの生成 — 自動検証とセキュリティスキャンを使用して、Amazon Elastic Kubernetes Service (Helm チャート) または Amazon Elastic Container Service (Terraform モジュール) のいずれかのデプロイインフラストラクチャを生成します。
- プライベート依存関係のサポート — 必要に応じて、AWS CodeArtifact リポジトリ (Maven、PyPI、npm) とプライベート Amazon ECR ベースイメージをビルドの依存関係ソースとして設定できます。

- 反復テストとカットオーバーのデプロイ — 検証用のテストインフラストラクチャをデプロイし、本番カットオーバー用に確定したインフラストラクチャをデプロイします。

コンテナ化の仕組み

AWS Transform は AI を活用したエージェントを使用して、チャットベースのワークフローでコンテナ化プロセスをガイドします。AWS Transform は、ソースコード分析、Docker イメージ生成、インフラストラクチャ作成などの特殊なタスクを調整します。ワークフローの重要なポイントで、先に進む前に出力を確認して承認します。

VMware 移行ジョブを作成して、ソースコードのコンテナ化にアクセスします。ジョブ内では、コンテナ化をスタンドアロンワークフローとして、またはウェブの移行戦略がコンテナ化に設定されている場合に end-to-end の移行の一部として実行できます。VMware 移行の詳細については、「」を参照してください [移行 \(VMware を含む\)](#)。

前提条件

作業を開始する前に、次の項目が揃っていることを確認してください。

- AWS 変換ワークスペース。ワークスペースの取得については、[「開始方法」](#)を参照してください。
- AWS CodeConnections からアクセスできる Git リポジトリ内のアプリケーションソースコード、またはアップロード用の zip ファイルとしてパッケージ化されているアプリケーションソースコード。個々のファイルは 1 GB を超えてはならず、すべてのソースコードの合計サイズは 8 GB を超えてはなりません。
- (オプション) コンテナイメージを発行するための Amazon ECR リポジトリ。公開する準備ができたなら、ワークフローの後半で Amazon ECR アクセスを設定できます。
- Amazon EKS デプロイの場合: 既存の Amazon EKS クラスター、または必要なインフラストラクチャを作成するためのアクセス許可。
- Amazon ECS デプロイの場合: Terraform を使用して Amazon ECS クラスター、サービス、および関連リソースを作成するアクセス許可 AWS CloudFormation、または AWS Cloud Development Kit (AWS CDK)。

コンテナ化ワークフロー

コンテナ化ワークフローは、次のステップで構成されます。AWS 変換エージェントは、チャットインターフェイスの各ステップをガイドします。

1. [the section called “ステップ 1: セキュリティに関する免責事項”](#) — セキュリティの免責事項を確認して受け入れます。
2. [the section called “ステップ 2: ソースコードのクローンを作成する”](#) — アプリケーションのソースコードを指定します。
3. [the section called “ステップ 3: コンテナ化”](#) — AI エージェントはコードを分析し、Docker アーティファクトを生成します。
4. [the section called “ステップ 4: アーティファクトを確認する”](#) — 生成されたアーティファクトを確認し、コード変更を承認します。
5. [the section called “ステップ 5: イメージを公開する”](#) — コンテナイメージを Amazon ECR に発行します。
6. [the section called “ステップ 6: IaC を生成する”](#) — Amazon EKS または Amazon ECS デプロイテンプレートを生成します。
7. [the section called “ステップ 7: テストインフラストラクチャをデプロイする”](#) — テストインフラストラクチャをデプロイして検証します。
8. [the section called “ステップ 8: テストインフラストラクチャをクリーンアップする”](#) — テストリソースをティアダウンします。
9. [the section called “ステップ 9: カットオーバーインフラストラクチャをデプロイする”](#) — 本番稼働用インフラストラクチャをデプロイします。

コンテナ化ジョブの開始

アプリケーションをコンテナ化するには、まず AWS Transform ワークスペースに VMware 移行ジョブを作成します。ジョブ内から、スタンドアロンのコンテナ化ワークフローを実行するか、コンテナ化を含むend-to-endの移行フローを実行するかを選択できます。

- スタンドアロンコンテナ化 — 完全な VMware 移行を実行せずにソースコードをコンテナ化します。インフラストラクチャの移行とは別にアプリケーションをコンテナ化する場合は、このオプションを選択します。
- コンテナ化によるEnd-to-endの移行 — 1 つ以上のウェーブに割り当てられたコンテナ化移行戦略を使用して、完全な VMware 移行ワークフローを実行します。コンテナ化ワークフローは、これ

らのウェブの移行の一部として実行されます。VMware 移行の詳細については、「」を参照してください[移行 \(VMware を含む\)](#)。

コンテナ化ジョブを開始するには

1. ワークスペースのランディングページで、ジョブの作成を選択します。
2. VMware 移行を選択します。
3. スタンドアロンのコンテナ化ワークフローを実行するか、コンテナ化を含むend-to-endの移行フローを実行するかを選択します。
4. AWS Transform は、 から始まるワークフローステップをガイドします[the section called “ステップ 1: セキュリティに関する免責事項”](#)。

コンテナ化のアクセス許可

コンテナ化ワークフロー中、AWS Transform は IAM ロールを にデプロイします。AWS アカウント は を使用してコンテナイメージを構築し、インフラストラクチャをデプロイします。ワークフローを続行する前に、このロールの作成を確認して承認するように求められます。

ロールは という名前でAWSTransformCodeBuildExecutionRole、AWS CloudFormation スタックを介してデプロイされます。これには、次の 管理ポリシーが含まれます。

基本ポリシー

コンテナ化ワークフローのコアアクセス許可を提供します。

- Amazon S3 — aws-transform-*バケット内のオブジェクトの読み取りと書き込み
- Amazon ECR — イメージの認証、プル (でタグ付けProject: atx-migration)、および でタグ付けされたリポジトリへのイメージのプッシュ CreatedBy: AWSTransform
- AWS CodeArtifact — でタグ付けされたリポジトリからの読み取りProject: atx-migration、リポジトリのリスト、認可トークンの取得、リポジトリエンドポイントの読み取りを行います。これらのアクセス許可は、ワークフロー中に設定されたパブリック依存関係解決とプライベート依存関係ソースの両方をサポートします。
- Amazon CloudWatch Logs — ロググループとストリームを作成し、 と Amazon ECS ロググループのログイベントを書き込む
- AWS KMS — キーの説明と復号 (Amazon S3 のサービス経由条件の範囲)
- Amazon EC2 — VPC 対応プロジェクトのネットワークインターフェイスの作成と管理

- AWS CodeConnections — 接続を使用して、設定された CodeConnections ARN を介してソースコードリポジトリにアクセスします。

ネットワークポリシー

デプロイされたアプリケーションのネットワークリソースを管理します。

- Elastic Load Balancing — ロードバランサー、ターゲットグループ、リスナー、およびルールを作成、記述、管理する (でタグ付けCreatedBy: AWSTransform)
- Route 53 — ホストゾーンと DNS レコードの作成と管理
- AWS クラウドマップ — サービス検出用の名前空間とサービスを作成および管理します (でタグ付けCreatedBy: AWSTransform)

ストレージポリシー

デプロイされたアプリケーションのストレージリソースを管理します。

- Amazon S3 — 暗号化、バージョニング、ライフサイクル、アクセスポリシーなどのバケットを作成および管理します。
- Amazon EFS — ファイルシステム、マウントターゲット、アクセスポイントの作成と管理 (でタグ付けCreatedBy: AWSTransform)
- Amazon EBS — ボリュームとスナップショットの作成と管理 (でタグ付けCreatedBy: AWSTransform)

AWS KMS ポリシー

暗号化キーオペレーションを管理します。

- キーの説明、エイリアスのリスト、およびキーポリシーの読み取り
- データキーの暗号化、復号、生成 (Amazon CloudWatch Logs、Amazon EFS、Amazon EC2、Amazon S3 のサービス経由条件の範囲)

Amazon ECS ポリシー

コンテナデプロイの Amazon Elastic Container Service リソースを管理します。

- クラスター、サービス、タスク定義の作成と管理 (でタグ付けCreatedBy: AWSTransform)
- タスク定義の登録と登録解除、タスクの実行
- Amazon ECS タスクとサービスに IAM ロールを渡す
- IAM ロール情報と ACM 証明書の読み取り
- Amazon EC2 VPCs、サブネット、セキュリティグループ、ネットワークインターフェイスについて説明する

Amazon EKS ポリシー

Kubernetes デプロイ用の Amazon Elastic Kubernetes Service リソースを管理します。

- Kubernetes API へのアクセス、クラスターの説明、アドオンのリスト
- Amazon EKS に IAM ロールを渡す

Note

すべてのアクセス許可は、AWS アカウント および に限定されます AWS リージョン。AWS Transform によって作成されたリソースは でタグ付けされCreatedBy: AWSTransform、書き込みオペレーションは必要に応じてこのタグが付いたリソースに制限されます。

ステップ 1: セキュリティの免責事項を確認する

コンテナ化ワークフローを開始する前に、AWS Transform は確認して承諾する必要があるセキュリティ上の免責事項を提示します。

免責事項の内容

コンテナ化プロセスでは、アプリケーションの依存関係 (npm、PyPI、Maven Central などのパブリックレジストリからのパッケージなど) をダウンロードするためにインターネットアクセスが必要です。セキュリティの免責事項は、以下を通知します。

- ビルド環境には、アプリケーションのビルド設定で指定された依存関係を解決およびダウンロードするためのアウトバウンドインターネットアクセスがあります。

- アプリケーションに必要な依存関係を確認し、それらが組織のセキュリティポリシーを満たしていることを確認する責任があります。
- 必要に応じて、後のステップでプライベート依存関係ソース (CodeArtifact リポジトリやプライベート Amazon ECR ベースイメージなど AWS) を設定して、パブリックレジストリからのダウンロードを回避できます。

必要な作業

セキュリティの免責事項を確認して承諾するには

1. AWS Transform が提示するセキュリティ上の免責事項をお読みください。
2. コンテナ化ワークフローを進めるには、免責事項を受け入れます。

免責事項を受け入れると、AWS Transform はソースコードを提供する次のステップに進みます。

Note

AWS CodeConnections と Amazon ECR のコネクタ設定は事前に不要になりました。AWS Transform は、必要に応じて、ワークフローの後半でこれらのサービスへのアクセスを設定するように求めます。たとえば、Git リポジトリからクローンを作成するときに CodeConnections を設定し、コンテナイメージを発行する準備ができたなら Amazon ECR アクセスを設定します。

ステップ 2: ソースコードのクローンを作成する

このステップでは、AWS 変換するアプリケーションのソースコードを指定します。Git リポジトリから、または zip ファイルをアップロードすることで、ソースコードを提供できます。

オプション 1: Git リポジトリ

Git リポジトリからソースコードを指定できます。AWS CodeConnections をまだ設定していない場合、AWS Transform はこの時点で CodeConnections ARN を指定するように求めます。リポジトリを指定するには、次の 2 つの方法があります。

- CSV ファイルをアップロードする — 列と を含む CSV ファイルを準備repo_nameしますbranch。は owner/repo-name形式repo_nameである必要があります。branch 列はオプションです。リポジトリのデフォルトブランチを使用するには、空白のままにします。例:

```
repo_name,branch
owner/my-app,main
owner/api-service,develop
owner/another-repo,
```

- チャットでリポジトリの詳細を指定する — CSV をアップロードするのではなく、チャットでリポジトリ名 (複数可) とブランチ (複数可) を直接 AWS 変換するように指示します。

AWS 変換は、各リポジトリが存在し、CodeConnections 設定からアクセスできることを検証し、リポジトリをクローンします。

Note

個々のファイルは 1 GB を超えてはならず、すべてのリポジトリの合計サイズは 8 GB を超えてはなりません。複数のリポジトリを同時に AWS 変換して、処理を高速化します。

Important

複数のリポジトリにまたがるアプリケーションは、Git クローンオプションではサポートされていません。アプリケーションで複数のリポジトリからのソースコードが必要な場合は、それらを zip ファイルとしてパッケージ化し、代わりに zip アップロードオプションを使用します。

オプション 2: Zip ファイルのアップロード

CodeConnections を設定していない場合、またはソースコードを直接アップロードする場合は、zip ファイルをアップロードできます。

ソースコードを zip ファイルとしてアップロードするには

1. プロンプトが表示されたら、アプリケーションごとに 1 つの zip ファイルをアップロードします。チャットにファイルをドラッグするか、アタッチメントボタンを使用できます。

2. 各 zip ファイルには、1 つのアプリケーションのソースコードが含まれている必要があります。
3. 個々のファイルは 1 GB を超えてはならず、すべてのアップロードの合計サイズは 8 GB を超えてはなりません。

オプション: プライベート依存関係を設定する

アプリケーションがプライベートレジストリのパッケージに依存している場合は、このステップ中にプライベート依存関係ソースを設定できます。AWS Transform は、次のプライベート依存関係タイプをサポートしています。

- AWS CodeArtifact リポジトリ — Maven、PyPI、または npm リポジトリを AWS CodeArtifact でホストするように設定します。AWS Transform は、アカウントで使用可能なリポジトリのリストを表示し、選択できるようにします。
- プライベート Amazon ECR ベースイメージ — Dockerfiles がプライベート Amazon ECR リポジトリからベースイメージを使用する場合は、ビルド環境がこれらのイメージをプルできるようにリポジトリの詳細を指定します。

プライベート依存関係を設定すると、AWS Transform はこれらのリソース AWS アカウント がホストされている に接続するように求めます。このコネクタ設定は、プライベート依存関係を使用する場合にのみ必要です。

コンテナ化設定の確認

ソースコードが利用可能になると、AWS Transform はコンテナ化の実行方法に関する具体的な指示があるかどうかを尋ねます。オプションで以下を指定できます。

- 新しい Dockerfile を生成するか、既存の Dockerfile を再利用するか。
- モノレポ内の特定のサービスを処理するか、すべてのサービスを処理するか。
- コンテナ化プロセスのその他の設定。

デフォルトは次のとおりです。

1. Amazon Linux 2023 ベースイメージ () を使用します `public.ecr.aws/amazonlinux/amazonlinux:latest`。
2. リポジトリ内のすべてのアプリケーションを処理します。
3. 既存の Dockerfile がある場合は再利用し、必要に応じてわずかな変更を加えます。

デフォルトを続行することを確認するか、特定の手順を指定できます。

ステップ 3: コンテナ化

このステップでは、AWS Transform AI エージェントはアプリケーションのソースコードを分析し、Docker アーティファクトを生成します。AWS Transform はアプリケーションの構造、依存関係、ランタイム要件を調べて、適切な Dockerfile と関連する設定ファイルを生成します。

コンテナ化中に何が起こるか

コンテナ化エージェントは、次のタスクを実行します。

1. ソースコード分析 — AWS 変換は、アプリケーションのプロジェクトファイル、依存関係、ビルド設定、ランタイム要件を調べます。AWS また、変換はハードコードされた値 (データベース IP アドレスや DNS 名など) を置き換え、環境変数を識別し、ボリュームを検出して、ログを標準出力にリダイレクトしようとします。
2. Dockerfile の生成 — 分析に基づいて、AWS Transform はアプリケーションに合わせた Dockerfile を生成します。
3. コンテナイメージのビルドとテスト — AWS Transform はコンテナイメージを構築し、Dockerfile が動作イメージを生成することを確認するテストを実行します。ビルドが失敗すると、AWS Transform は Dockerfile で自動的に反復します。
4. セキュリティ分析 — AWS 変換は、生成されたアーティファクトをスキャンして、認証情報、API キー、その他の機密データなどのハードコードされた値がないか確認し、レビューのために検出結果にフラグを付けます。

複数のリポジトリまたはモノレポに複数のサービスを提供した場合、は各サービスを並行して処理します。

サポートされるアプリケーションタイプ

AWS Transform は、次のアプリケーションタイプのコンテナ化をサポートしています。

- .NET 7 以降 — バージョン 7 以降で実行されている .NET アプリケーションのコンテナ化を最適化しました。
- 一般的なアプリケーション — Java、Python、Node.js、Go など、任意の言語またはフレームワークで構築されたアプリケーションのコンテナ化。

- サポートされていないアプリケーション — Windows、Xcode、または独自のビルド環境はサポートされていません。

必要な作業

このステップは自動的に実行されます。AWS Transform は、各アプリケーションを分析およびコンテナ化する際に進行状況の更新を表示します。AWS Transform で問題が発生した場合は、明確化や追加情報を求められることがあります。

コンテナ化が完了すると、AWS Transform は次のステップに移行し、生成されたアーティファクトを確認します。出力は Artifact Store に保存されます。

ステップ 4: Docker アーティファクトとコード変更を確認する

このステップでは、コンテナ化中に AWS Transform が生成した Docker アーティファクトとコード変更を確認します。

Git リポジトリからの変更の確認

Git リポジトリからソースコードを指定した場合、AWS Transform はコンテナ化の変更を各リポジトリの新しいブランチにコミットします。複数のリポジトリがある場合、AWS Transform はすべての変更を一括レビュー用のバッチとして表示します。

変更を確認して承認するには

1. AWS Transform が提供するコード差分リンクを確認します。各差分には、Dockerfiles、設定ファイル、ソースコードの変更など、追加または変更されたすべてのファイルが表示されます。
2. AWS Transform が承認ダイアログを表示したら、変更を承認または拒否します。複数のリポジトリの場合、1 回のバッチオペレーションですべての変更を承認します。
3. 承認すると、AWS Transform はリポジトリ内の新しいブランチに変更をプッシュします。
4. 次のステップに進む準備ができていることを確認します。

zip アップロードからの変更の確認

ソースコードを zip ファイルとしてアップロードした場合、AWS Transform はコンテナ化されたアーティファクトのダウンロードリンクを提供します。生成された Dockerfile とソースコードの変更を含むアーティファクトをダウンロードして確認します。

変更のリクエスト

生成されたアーティファクトに変更が必要な場合は、AWS 変換が更新された手順でコンテナ化ステップを再起動し、レビュー用の新しいアーティファクトを生成するようにフィードバックを提供できます。

ステップ 5: イメージを公開する

このステップでは、AWS Transform はコンテナイメージを Amazon Elastic Container Registry に発行します。Amazon ECR アクセスをまだ設定していない場合、AWS Transform はこの時点で Amazon ECR リポジトリの詳細を指定するように求めます。

公開中に何が起こるか

AWS Transform は、コンテナイメージごとに次のタスクを実行します。

1. コネクタ設定 — Amazon ECR アクセスをまだ設定していない場合、AWS Transform は Amazon ECR リポジトリ ARN を指定して に接続するコネクタフォームを表示します AWS アカウント。
2. 承認リクエスト — AWS 変換は発行するイメージのリストを表示し、承認をリクエストします。発行オペレーションを承認または拒否できます。
3. イメージプッシュ — 承認後、AWS Transform は を使用してコンテナイメージを Amazon ECR リポジトリに公開します。複数のサービスがある場合、AWS Transform は 1 回のバッチオペレーションですべてのイメージを発行します。
4. 脆弱性スキャン — Amazon ECR は、公開されたイメージの既知の脆弱性を自動的にスキャンします。AWS Transform はスキャン結果をレポートします。より包括的な分析を行うには、AWS ECR 拡張スキャンと AWS GuardDuty Runtime Monitoring を有効にします。

Note

イメージ発行は、 を使用して非同期的に実行されます。公開中にセッションが中断された場合、AWS Transform は再接続時にオペレーションを自動的に復旧します。

必要な作業

- 公開するイメージのリストを確認します。

- 発行オペレーションを承認または拒否します。拒否すると、AWS Transform はイメージの公開をスキップし、イメージを公開せずに次のステップに進むことができます。
- 公開が完了したら、脆弱性スキャンの結果を確認します。

公開後、AWS Transform はインフラストラクチャデプロイステップで使用されるイメージ URIs を提供します。

ステップ 6: Infrastructure as Code を生成する

このステップでは、AWS Transform はコンテナ化されたアプリケーションをデプロイするための AWS インフラストラクチャテンプレートを生成します。ターゲットプラットフォームとして、Amazon Elastic Kubernetes Service と Amazon Elastic Container Service のいずれかを選択します。

Amazon Elastic Kubernetes Service (Kubernetes)

Amazon EKS を選択した場合、AWS Transform はアプリケーションの Kubernetes マニフェストを含む Helm チャートを生成します。生成されたテンプレートには以下が含まれます。

- デプロイとサービス
- 受信ルール
- ConfigMaps および SecretProviderClass リソース
- サービスアカウント
- 永続ボリュームクレーム (EBS および EFS)
- Route 53 DNS レコード
- ヘルスチェック設定

オプション: CloudWatch ログ記録

AWS Transform は、Amazon EKS デプロイに CloudWatch ログ記録を含めるかどうかを尋ねます。オプトインすると、AWS Transform はポッドログを収集して CloudWatch Logs に配信する Fluent Bit DaemonSet 設定を生成します。

CloudWatch ログ記録を選択した場合、クラスター管理者はデプロイ前に次の前提条件を満たす必要があります。

1. クラスターにamazon-cloudwatch名前空間を作成します。
2. CloudWatch Logs の書き込み権限を持つ Fluent Bit サービスアカウントの IAM ロールを作成します。デプロイ中にロール ARN の入力を求められます。

Amazon Elastic Container Service (Terraform)

Amazon ECS を選択した場合、AWS Transform はアプリケーションをデプロイするための Terraform モジュールを生成します。生成されたテンプレートには以下が含まれます。

- ECS クラスター
- タスク定義を含む ECS サービス
- アクセスログ記録を使用する Application Load Balancer
- 永続的ストレージ用の Amazon EFS
- サービス検出用のプライベートホストゾーン
- ヘルスチェック設定

自動検証とセキュリティスキャン

AWS Transform は、生成されたインフラストラクチャテンプレートを自動的に検証してから提示します。検証には次のものが含まれます。

- Terraform `terraform validate` の場合: `terraform fmt` 構文と設定の正確性を確認するには、`terraform fmt` を使用します。
- Helm の場合: 生成されたマニフェストの形式が適切であることを確認するグラフ検証。
- Checkov を使用したセキュリティスキャンにより、生成されたインフラストラクチャコード内の潜在的なセキュリティ設定ミスを特定します。

AWS Transform は、検証の問題を報告し、続行する前に解決します。

公開されたイメージなしで IaC を生成する

前のステップでコンテナイメージを公開していない場合でも、インフラストラクチャテンプレートを生成できます。生成されたテンプレートはプレースホルダーイメージ URIs、後で実際のイメージの場所で更新できます。

必要な作業

インフラストラクチャテンプレートを生成するには

1. プロンプトが表示されたら、ターゲットプラットフォーム Amazon EKS または Amazon ECS を選択します。
2. Amazon EKS を選択した場合、AWS Transform は CloudWatch ログ記録を含めるかどうかを尋ねます。「はい」または「いいえ」と回答します。
3. プロンプトが表示されたら、アプリケーションのヘルスチェックエンドポイントの詳細 (/healthパスなど) を指定します。
4. AWS Transform はインフラストラクチャテンプレートを生成して検証します。このステップは自動的に実行され、AWS Transform は検証やセキュリティスキャンの結果を含む進行状況の更新を表示します。
5. Transform がテンプレート AWS を表示するときに、生成されたテンプレートを確認します。

ステップ 7: テストインフラストラクチャをデプロイする

このステップでは、AWS Transform はテストインフラストラクチャをデプロイし、本番稼働のカットオーバー前にコンテナ化されたアプリケーションを検証できるようにします。

テストデプロイ中に何が起こるか

AWS Transform は、前のステップで生成されたインフラストラクチャテンプレートを にデプロイします AWS アカウント。Amazon EKS デプロイでは、Helm が使用されます。Amazon ECS デプロイでは、Terraform が使用されます。

デプロイする前に、AWS Transform はリソース名、レプリカ数、環境固有の設定などのデプロイパラメータを設定できるインフラストラクチャ値フォームを表示します。AWS Transform は、作成されるリソースを示すデプロイプレビューも生成するため、承認する前に変更を確認できます。

必要な作業

テストインフラストラクチャをデプロイして検証するには

1. デプロイ設定でインフラストラクチャ値フォームに入力します。フォームには、クラスター名、名前空間、レプリカ数、その他の環境固有の設定などのパラメータが含まれます。
2. 作成または変更されるリソースを示すデプロイプレビューを確認します。

3. AWS Transform がリクエストしたときにインフラストラクチャのデプロイを承認します。
4. デプロイが完了するまで待ちます。AWS Transform はデプロイ結果を表示します。
5. デプロイしたアプリケーションをテストして、コンテナ内で正しく実行され、インフラストラクチャが要件を満たしていることを確認します。
6. デプロイに満足しているかどうか、またはインフラストラクチャを変更して再デプロイするかどうかを AWS 変換に指示します。

インフラストラクチャでの反復処理

インフラストラクチャテンプレートを変更する必要がある場合は、更新されたテンプレートをアップロードして再デプロイできます。

1. IaC を変更する AWS 変換を指示します。
2. 変更されたインフラストラクチャテンプレートを含む zip ファイルをアップロードします。
3. AWS Transform はテンプレートを置き換え、増分更新 (Terraform 適用または Helm アップグレード) を実行します。
4. 更新されたデプロイ結果を確認します。

このサイクルは、デプロイに満足するまで繰り返すことができます。

セッション復旧

インフラストラクチャデプロイは、を使用して非同期的に実行されます。デプロイ中にセッションが中断された場合、AWS Transform は再接続するとオペレーションを自動的に復旧し、デプロイ結果を表示します。

ステップ 8: テストインフラストラクチャをクリーンアップする

検証が完了すると、AWS Transform は不要なコストを回避するためにテストインフラストラクチャを破棄します。

必要な作業

テストインフラストラクチャをクリーンアップするには

1. AWS 変換は、テストインフラストラクチャが切断されることを通知します。

2. AWS Transform がリクエストしたときに、破棄オペレーションを承認します。
3. クリーンアップが完了するまで待ちます。AWS Transform は、すべてのテストリソースが削除されると結果を表示します。

Amazon ECS デプロイの場合、このステップでは、コンピューティングリソース、ロードバランサー、ストレージ、ネットワークコンポーネントなど、テストデプロイ中に作成されたすべてのリソースを削除します。

Amazon EKS デプロイの場合、クリーンアップは特定のプロジェクトに対するラベルによってスコープされます。このコンテナ化ワークフローにタグ付けされたリソースのみが削除されます。同じ Amazon EKS クラスターで実行されている他のワークロードは影響を受けません。

ステップ 9: カットオーバーインフラストラクチャをデプロイする

この最後のステップでは、AWS Transform は本番稼働用の確定済みインフラストラクチャをデプロイします。カットオーバーデプロイでは、テストデプロイステップで検証したのと同じインフラストラクチャテンプレートを使用します。

必要な作業

カットオーバーインフラストラクチャをデプロイするには

1. 本番デプロイ設定でインフラストラクチャ値フォームに入力します。テストデプロイからパラメータを調整することも、同じ値を使用することもできます。
2. 作成される本番稼働用リソースを示すデプロイプレビューを確認します。
3. AWS Transform がリクエストしたときに本番デプロイを承認します。
4. デプロイが完了するまで待ちます。AWS Transform はカットオーバーデプロイの結果を表示します。
5. アプリケーションが本番環境で正しく実行されていることを確認します。

Important

生成されたインフラストラクチャ設定をチームで確認し、組織のセキュリティ、コンプライアンス、ビジネス要件を満たしていることを確認します。AWS Transform は自動設定レコメンデーションを提供しますが、カットオーバーデプロイに進む前に、ニーズに合わせて設定を検証および調整する責任があります。

セッション復旧

本番稼働用デプロイは、 を使用して非同期的に実行されます。デプロイ中にセッションが中断された場合、AWS Transform は再接続するとオペレーションを自動的に復旧し、デプロイ結果を表示します。

メインフレームアプリケーションのモダナイゼーション

AWS 変換は、レガシーメインフレームアプリケーションのモダナイゼーションを加速するように設計されています。メインフレームコードベースの分析をオーケストレーションし、ドキュメントを生成し、ビジネスロジックを抽出し、モノリシック構造を分解し、レガシーコードを変換し、必要に応じて人間による入力 (HITL) でジャーニー全体を管理します。メインフレームアプリケーションをモダナイズおよび移行するための AWS トランスフォーメーション機能により、トランスフォーメーションプロセス全体でビジネスクリティカルなロジックを維持しながら、重要なメインフレームアプリケーションをより迅速にモダナイズできます。

トピック

- [機能と主要機能](#)
- [大まかなウォークスルー](#)
- [ヒューマンインザループ \(HITL\)](#)
- [メインフレームアプリケーションの変換でサポートされるファイルタイプ](#)
- [AWS Transform メインフレームでサポートされているリージョンとクォータ](#)
- [メインフレームのモダナイゼーションジャーニーの概要](#)
- [メインフレームアプリケーションの変換](#)
- [リファクタリング後にモダナイズされたアプリケーションを構築してデプロイする](#)

機能と主要機能

AWS Transform は、メインフレームのモダナイゼーションに次の機能を提供します。

- COBOL (Common Business-Oriented Language) および PL/I (プログラミング言語 1) で記述された z/OS メインフレームアプリケーションのモダナイゼーションを、関連する JCL (Job Control Language)、CICS (Customer Information Control System) トランザクション、BMS (Basic Mapping Support) 画面、Db2 データベース、および VSAM (Virtual Storage Access Method) データファイルでサポートします。
- PSAM (Presentation Service Access Method)、日本語文字セット、および NDB (Network Data Base) データファイルを使用した富士通 GS21 メインフレームアプリケーションのリファクタリングをサポートします。
- 目標駆動型の推論、分析、分解、計画、ドキュメント生成、コードリファクタリングを実行します。

- COBOL ベースのメインフレームワークロードを最新のクラウド最適化 Java アプリケーションに自動的にリファクタリングします。
- 分析、ドキュメント、分解、計画、コードリファクタリングを実行する基盤となるツールとシームレスに調整および統合します。
- ready-to-use Infrastructure as Code (IaC) テンプレートを提供することで、モダナイズされたメインフレームアプリケーションのクラウド環境をセットアップするのに役立ちます。

大まかなワークスルー

メインフレームアプリケーションをモダナイズおよび移行するための AWS Transform の大まかなワークスルーを次に示します。

1. AWS 変換でチャットを開始し、目標を入力します。
2. 目的に基づいて、AWS Transform はモダナイゼーション計画を提案します。大まかな目標を中間ステップに分割します。
3. 指定した目標に応じて、AWS Transform は次のことができます。
 - コードベースを評価して再考する
 - 目標に基づいてカスタムジョブを作成する

その過程で、AWS Transform はタスクを実行するためにユーザーに情報をリクエストすることがあります。

Note

メインフレームリファクタリング機能の AWS 変換の詳細については、メインフレームリファクタリングの AWS 変換ドキュメントの [「プロジェクトを作成する」](#) を参照してください。

ヒューマンインザループ (HITL)

メインフレームアプリケーションの変換全体を通して、AWS 変換ウェブエクスペリエンスを通じて変換タスクの進行状況とステータスをモニタリングできます。

AWS Transform は、以下のシナリオで追加情報を収集します。

- タスクを実行するために追加情報が必要な場合。
- 中間アーティファクト (ドメイン分解やウェーブプランニングなど) の承認が必要な場合。
- Transform が自動解決 AWS できない問題が発生した場合。

メインフレームアプリケーションの変換でサポートされるファイルタイプ

z/OS でサポートされているファイルタイプは次のとおりです。

- COBOL アーティファクトと関連する CPY (コピーブック)
- PL/I ソースファイル
- JCL (ジョブコントロール言語) と JCL プロシージャ (PROC)
- CICS システム定義 (CSD)
- BMS (基本マッピングのサポート)
- Db2 データベース
- VSAM (仮想ストレージアクセス方法)
- IMS TM (Transaction Manager)

Fujitsu GS21 でサポートされているファイルタイプは次のとおりです。

- PSAM (プレゼンテーションサービスアクセス方法)
- ADL (AIM 定義言語)
- NDB (ネットワークデータベース)

Fujitsu GS21 の詳細については、AWS メインフレーム移行のための変換ガイドの以下のトピックを参照してください。

- [GS21](#)
- [キャプチャと再生 - GS21 ターミナル](#)
- [Mainframe、AS400、Open VMS、GS21](#)

AWS Transform メインフレームでサポートされているリージョンとクォータ

サポートされているリージョンのリストについては、「」を参照してください[でサポートされているリージョン AWS Transform](#)。

Note

データは、AWS 変換を使用するリージョンとは異なるリージョンで処理される場合があります。クロスリージョン処理の詳細については、「」を参照してください[the section called “クロスリージョン処理”](#)。

クォータの制限については、「」を参照してください[クォータ](#)。

メインフレームのモダナイゼーションジャーニーの概要

メインフレームアプリケーションを にモダナイズするには、通常 AWS 、(a) 評価、(b) 動員、(c) 移行とモダナイズ、(d) 運用、最適化、イノベーションの 4 つのフェーズがあります。これらの各フェーズは、モダナイゼーションジャーニーをサポートするために、目標と関連するアクティビティで詳細に説明されています。

フェーズ

- [フェーズ 1: 評価](#)
- [フェーズ 2: 動員](#)
- [フェーズ 3: 移行とモダナイズ](#)
- [フェーズ 4: 運用、最適化、イノベーション](#)

フェーズ 1: 評価

目標: モダナイゼーションの準備状況を理解し、最初のモダナイゼーション計画を立てる。

AWS Transform を使用すると、メインフレームアプリケーション、インフラストラクチャ、運用を評価して、ビジネスニーズに適したモダナイゼーションアプローチを決定できます。

Note

各フェーズには、メインフレームアプリケーションをモダナイズするときに実行できるすべてのアクティビティのリストが含まれています。ユースケースや目標に合わないアクティビティの一部はスキップできます。

アクティビティ

- モダナイゼーションに関する非公式な会話から始め、適切な人材を会議室に集めて機会認定に取り組む
- 機会認定を決定する
- アプリケーション依存関係マッピングプランとアプリケーションアーキテクチャを使用して、初回コールプレゼンテーションを行う
- 移行中のインフラストラクチャコスト、移行作業、ポストプロダクションコスト、プロジェクト期間を含む ROM (大まかな桁数) を決定する
- 迅速なポートフォリオ検出:
 - 実行パターンとツールガイダンスの評価
 - ソースコード分析を行う
 - 概念実証 (POC) 候補を特定する
- 技術的な有効化トレーニングを作成します。
 - 没入日数
 - 一般的なトピックを含むワークショップ
- 大規模なメインフレームモダナイゼーションプロジェクトまたは新規 AWS 顧客の移行準備評価 (MRA) レポートを作成する
- Mobilize フェーズの作業明細書を作成する

フェーズ 2: 動員

目標: 概念実証またはパイロットモダナイゼーションを成功させることで、基盤を構築し、アプローチを検証する。

動員フェーズでは、メインフレームのモダナイゼーションを大規模にサポートするために必要なフレームワーク、ツール、プロセスを確立できます。

アクティビティ

- パイロット版と特定された POC を使用してモダナイゼーション計画を開始します。
 - パイロットを分析し、モダナイゼーション用のコードをセットアップする
 - パイロットモダナイズされたデータを構築してテストする
 - 結果を配信し、パイロット版を確認して、大規模なデータモダナイゼーションのための詳細なソリューションを作成する
- プラットフォームの場合:
 - セキュリティ、コンプライアンス、モニタリング計画を作成する
 - AWS メインフレームランディングゾーンとアカウント構造を使用してランディングゾーン環境を設定する
 - アーキテクチャインフラストラクチャ、アプリケーション、統合などを決定する
- 人とプロセスについて考えます。
 - Cloud Center of Excellence は優れたリソースです
 - オペレーションモデル
- さまざまなアプリケーションシナリオの包括的なポートフォリオ分析を作成します。
 - 高レベルのドキュメントを含む包括的な検出
 - 分解計画 (すべてのアプリケーション用)
 - すべてのアプリケーションのパターンとツールのガイダンス
 - 初期のモダナイゼーションと移行計画を立てる
- パイロット、プラットフォーム、人材とプロセス、包括的なポートフォリオ分析を考慮した詳細なソリューションを作成する
- ビジネスケースを決定する
- 次のフェーズの作業明細書のソリューションでパイロットを成功させる

フェーズ 3: 移行とモダナイズ

目標: 計画されたタイムラインと予算内にモダナイゼーションと移行を完了する。

このフェーズでは、メインフレームアプリケーションとデータを に移行してモダナイズできます AWS。

アクティビティ

- 本番環境対応のアプリケーションコードにリソースを割り当てる
- 既存のドキュメントを使用するか、対象分野の専門家 (SMEs) にインタビューして、設計を確認して分析する
- テストシナリオを定義する
- 適切なランタイム、データベース、サードパーティーソフトウェアを使用してモダナイゼーション環境をセットアップする
- ソースコードとデータを移行してモダナイズする:
 - アプリケーションコードベースのビジネスドメインに基づいて、アプリケーションを繰り返しモダナイズできます。
 - モダナイゼーションワークロードをサポートするために、必要に応じてモダナイゼーション環境をスケールする
- CI/CD (継続的インテグレーション、継続的デリバリー) パイプラインを確立する
- 選択したワークロードを移行してモダナイゼーションを開始します。
- アプリケーションプログラムを移管し、必要に応じて環境スケーリングを調整します。
 - 必要に応じて環境をスケールアップまたはスケールダウンできます
 - コードサニティテストを実行して作業パッケージをモダナイズする
 - 必要に応じてソースコードと再同期する
- 包括的なテストを実行し、移行を検証します。
 - 初期状態データの収集
 - メインフレームにテストシナリオを記録する
 - 環境、アプリケーション、データのステージング
 - クラウドでテストシナリオを実行する
 - ソーステスト結果とターゲットテスト結果を比較する
 - 自動テストシナリオ実行パイプラインを構築する
 - KPIs とレポートの生成
- 本番稼働用セットアップのデータを検証する
- 次の方法で進行状況を検証します。
 - 統合とシステムテストの実行
 - パフォーマンステストの実行
 - ユーザー承認テストの実行

- HA/DR テストの実行
- アプリケーション所有者とのロールアウト計画を定義し、トレーニングと知識移転の計画を作成する
- カットオーバーを検討し、本番稼働用のカットオーバーを実行する
- 運用、モニタリング、機能計画、アプリケーション機能の保証など、本番稼働後のアクティビティに対する継続的なサポートを確立する

フェーズ 4: 運用、最適化、イノベーション

目標: 運用上の優秀性とモダナイズされたアプリケーションの継続的な改善をサポートする。

モダナイゼーション後、イノベーションのために AWS サービスを活用しながら、アプリケーションのパフォーマンス、セキュリティ、コストを最適化できます。

アクティビティ

- 以下についてアプリケーションをモニタリングします。
 - パフォーマンスメトリクス
 - セキュリティプラクティス
 - コンプライアンスガイドライン
 - コスト
- ログ記録の詳細
- アプリケーションオペレーションを管理します。
 - インフラストラクチャ管理
 - トランザクションとジョブ管理
 - ランタイム環境のサポート
- アプリケーションの最適化とモダナイゼーション
- アプリケーションの開発とメンテナンス:
 - コーディング IDE
 - 構築
 - 統合
 - テスト
 - デプロイメント

メインフレームアプリケーションの変換

AWS トランスフォームは、メインフレームのモダナイゼーションアプリケーションのトランスフォーメーションを加速します。このトピックでは、使用可能な機能について説明します。

トピック

- [前提条件: S3 でプロジェクト入力を準備する](#)
- [メインフレームソースアーティファクトコレクション](#)
- [サインインしてジョブを作成する](#)
- [変換の進行状況の追跡](#)
- [コネクタのセットアップ](#)
- [ジョブプランの評価と再考](#)
- [ジョブプランを再考する](#)
- [カスタムジョブプラン](#)

前提条件: S3 でプロジェクト入力を準備する

AWS 変換は、複雑なメインフレームコードベースを処理できます。コードベースを使用するには、S3 ロケーションにすべてのアセットがあることを確認してください。

主要なプロジェクト入力:

- ソースコード: メインフレームのソースコードファイルを S3 にアップロードする必要があります。これには、COBOL プログラム、JCL スクリプト、コピーブック、およびその他の関連するソースファイルが含まれます。
- データファイル: メインフレームアプリケーションが使用する VSAM ファイルやその他のデータファイルがある場合は、S3 にアップロードする必要があります。
- 設定ファイル: メインフレーム環境に固有の設定ファイルを含めます。

その他のプロジェクト入力

- System Management Facility (SMF) レコード: 該当する場合は、ソースコードが保存されている S3 バケットの新しいフォルダに SMF レコードをアップロードします。これらのレコードは .zip ファイル形式である必要があります。

- **フォーマット:** 技術ドキュメントの生成では、オプションの 設定ファイル を活用して、ヘッダー、フッター、ロゴ、カスタマイズされた情報など、必要な形式と標準に沿った PDF ドキュメントを生成できます。
- **用語集:** AWS Transform AWS Transform は、Generative AI による自動化を活用して、ドキュメントの生成とビジネスルールの抽出を行います。zip ファイルのルートディレクトリに重要な略語と用語に関する情報を含む用語集 CSV ファイルを含めると、生成されたドキュメントの品質が向上します。
- **テストデータ:** 利用可能な場合は、モダナイズされたアプリケーションの検証に使用できるテストデータセットをアップロードします。このデータは、ソースコードが保存されている S3 バケットの新しいフォルダに保存する必要があります。

プロジェクト入力の詳細については、[こちら](#)を参照してください。

メインフレームソースアーティファクトコレクション

このトピックでは、AWS Transform を使用してメインフレームのモダナイゼーションイニシアチブをサポートするために、メインフレームのソースコードと関連アーティファクトを収集するプロセスの概要を説明します。分析、移行計画、実装を成功させるには、ソースアーティファクトの適切な収集と準備が不可欠です。

ソースコードインベントリ

アプリケーションのソースコードタイプ

次のアプリケーションソースコードタイプがサポートされています。

- COBOL プログラム
- JCL スクリプト
- アセンブラプログラム
- PL/I プログラム
- CICS トランザクション
- 自然プログラム
- REXX スクリプト
- Easytrieve プログラム
- コピーブックとを含む

• Proc ライブラリ

データベースアセット

次のデータベースアセットがサポートされています。

- DB2 DCLGEN スクリプトと DDL スクリプト
- IMS データベース定義 (PSBsと DBDs)
- VSAM ファイル定義
- 物理データベーススキーマと論理データベーススキーマ
- データベースの手順とトリガー

ソースコードファイルの拡張子

拡張機能が不明な場合は、空白または .txt のままにすると、AWS 変換によって自動的に分類されます。

次の表に、メインフレームソースアーティファクトの一般的なファイル拡張子を示します。

メインフレームソースコードファイルの拡張子

言語/タイプ	一般的な拡張機能	説明
COBOL	.cbl、.cob、.cobol	COBOL ソースプログラム
JCL	.jcl	ジョブコントロール言語スクリプト
アセンブラ	.asm	アセンブラ言語プログラム
PL/I	.pl1	PL/I 言語プログラム
CICS 定義	.csd	CICS システム定義 (CSD)
自然	.nat	自然プログラム
REXX	.rex、.rexx	REXX スクリプト
Easytrieve	.ezt	Easytrieve レポートプログラム
コピーブック	.cpy	COBOL コピーブック

言語/タイプ	一般的な拡張機能	説明
BMS	.bms	基本的なマッピングのサポート (BMS)
PL/I コピーブック	.pl1_copy	PL/I コピーブック
Db2 DCLGEN	.dcl	Db2 宣言ジェネレーター
Db2 定義	.sql、.ddl	Db2 データベース定義
IMS 定義	.ims	IMS リソース定義データセット (IMS ステージ 1)
MFS	.mfs	IMS MFS (メッセージ形式サービス)
PSB	.psb	IMS PSB (プログラム仕様ブロック)
DBD	.dbd	IMS データベースの定義
JCL Proc	.prc、.proc	JCL プロシージャライブラリ
JCL に含まれるもの	.inc	JCL ファイルを含める
マクロ	.mac	アセンブルマクロ
コントロールカード	.ctl	JCL ユーティリティコントロールカード

収集方法

ソースコードの抽出

メインフレーム PDS ライブラリからのインタラクティブなソースコード抽出には、セキュア FTP クライアント (WinSCP/FileZilla) を使用します。

DB2 スキーマの抽出

DB2 LUW (Linux、UNIX、Windows) クライアントソフトウェアには、インストールパッケージの一部として db2look ユーティリティが含まれています。db2look ユーティリティは、DB2 データベースから DDL (データ定義言語) ステートメントを抽出します。

次のコマンドを実行します。

```
db2look -d DATABASE_NAME -i userid -w password -a -e -m -l -x -f -createdb -printdbcfg  
-o DATABASE_NAME.sql
```

次のリストでは、パラメータについて説明します。

- -d *DATABASE_NAME*: スキーマを抽出するデータベースの名前を指定します。
- -i *userid*: データベース認証のユーザー ID
- -w *password*: データベース認証のパスワード
- -a: 認可情報 (グラント) を抽出します。
- -e: すべてのデータベースオブジェクト (テーブル、ビュー、インデックスなど) を抽出します。
- -m: 統計と物理特性を抽出します
- -l: テーブルと列のコメントを抽出します
- -x: 説明テーブルを抽出します
- -f: 読みやすくするために出力をフォーマットします
- -createdb: データベース作成ステートメントが含まれます
- -printdbcfg: データベース設定パラメータが含まれます
- -o *DATABASE_NAME*.sql: 出力ファイル名を指定します。

メインフレームからのカタログ情報の抽出

IDCAMS の LISTCAT コマンドは、メインフレーム上のデータセットに関するカタログ情報を取得するためのユーティリティです。これは、メインフレームのモダライゼーションプロジェクトでデータセットをインベントリし、その特性を理解するのに役立ちます。

```
//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
```

```
/*
```

CICS 定義の抽出

IBM が提供する DFHCSDUP ユーティリティは、CICS システム定義 (CSD) ファイルから CICS リソース定義を抽出できます。

次の JCL を使用して、DFHCSD から既存の定義をアンロードします。

```
//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
/*
```

IMS 定義の抽出

IMS 定義を抽出するには、DFSURDD0 ユーティリティを使用して IMS トランザクションファイルを生成できます。このユーティリティは、移行目的で利用できるステージ 1 マクロステートメントを作成します。現在サポートされているのは IMS ステージ 1 のみです。

次の JCL を使用して IMS 定義を抽出します。

```
//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  
/*
```

CA-7 スケジューラ情報の抽出

CA-7 ジョブスケジュール情報は LJOB レポートに含まれています。LIST=NODD パラメータで LJOB ユーティリティを使用して、メインフレームでレポートを生成します。レポートには .ca7 拡張子が必要です。各レポートには、列 1 に ANSI キャリッジコントロール文字が必要です。LJOB ユーティリティの実行に使用される JCL で DCB=RECFM=FBA (または FA) を指定することで、列 1 にキャリッジコントロール文字を含むレポートを作成できます。

例: LJOB, JOB=ZBN*, LIST=ALL

詳細については、[Broadcom のドキュメント](#)を参照してください。

SMF データの抽出

System Management Facility (SMF) レコードは、AWS Transform がバッチジョブと CICS トランザクションの使用状況を分析するために使用されるアクティビティメトリクスを提供します。次の SMF レコードタイプがサポートされています。14、15、30 (サブタイプ 5)、64、102、110 (データクラス 1、3、4 を持つサブタイプ 1)。

SMF レコードの形式要件:

- レコードタイプ 30 および 110 を含める必要があります (最低)
- RDW (Record Descriptor Word) バイトを含む raw バイナリ EBCDIC 形式である必要があります
- 次のいずれかとして指定する必要があります。
 - .zip 形式で圧縮された最大 600 MB のファイル
 - .tar.gz または .gz 形式で圧縮された最大 5 GB のファイル
- Amazon S3 のソースコードとは別のフォルダに保存する必要があります

最低 13 か月の SMF レコードを提供して、短い時間枠で見逃す可能性のある年間バッチサイクルと季節的なパターンをキャプチャします。

オプション 1: SMF レコードを抽出およびソートする

次のサンプル JCL を使用して、SMF レコードを抽出およびソートします。

```
//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
/*
```

前述の JCL で次の値をカスタマイズします。

1. ジョブカード: アカウント、クラス、msgclass を更新する
2. DUMPIN: SMF データセットを指す
3. DUMPOUT: 出力データセット名と HLQ を指定する
4. タイプ: レコードタイプ (14、15、30、110 など) を選択する
5. DATE: yyyy/ddd の形式 (例: 2026/055、2026/056)
6. SPACE: 予想されるデータ量に基づいて調整する
7. 不要な場合は SORT ステップを削除する

オプション 2: DFSORT を使用して SMF データを抽出する

DFSORT を使用して、SMF の履歴データを統合してレポートできます。これは、SMF レコードを日別または週別に GDG に保存する場合に役立ちます。次のジョブでは、DFSORT を使用して SORTIN ファイルから SMF レコードを統合し、INCLUDE COND パラメータで指定されたレコードに基づいてフィルタリングします。

```
//EXTRSMF JOB 'EXTRACT SMF RECORDS',CLASS=S,MSGCLASS=H,
// MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID,TIME=1440
//SORTJES2 EXEC PGM=DFSORT,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')
```

RDW データを保持しながらデータをダウンロードするには、FTP サーバーを使用して抽出されたファイルを受信します。RDW がデータ転送に含まれるように、データがバイナリ形式で転送され、RDW オプションが使用されていることを確認します。次のジョブは、このファイルを転送するためのサンプル FTP ジョブを示しています。

```
//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
```

SCRT レポートのアップロード

また、Sub-Capacity Reporting Tool (SCRT) レポートを AWS Transform にアップロードすることもできます。

アプリケーションインベントリのアップロード

すべてのアーティファクトを収集したら、次の手順を実行します。

1. アーティファクトタイプ (/cobol、/jcl、/copybooks、/db2、/ims、/cics、/ca7 など) でファイルをサブフォルダに整理します。

2. 最上位フォルダを 1 つの .zip ファイルに圧縮します。
3. AWS 変換ワークスペースに接続されている Amazon S3 バケットに .zip ファイルをアップロードします。

Note

SMF レコードは、Amazon S3 バケット内のソースコードとは別のフォルダに配置し、.tar.gz または .gz 形式である必要があります。

Amazon S3 バケットを AWS 変換に接続する詳細な手順については、「」を参照してください [the section called “コネクタのセットアップ”](#)。

サインインしてジョブを作成する

AWS Transform ウェブエクスペリエンスにサインインするには、ドキュメントの [AWS 変換の開始方法](#) セクションのすべての手順に従ってください。

ジョブを作成して開始するには、[「プロジェクトを開始する」のステップに従います。](#)

ワークスペースの作成: ジョブ、共同作業、および関連するアーティファクトが保存されるワークスペースに名前を付け、記述します。

ジョブの作成: 事前設定されたジョブプランから選択してジョブを作成するか、サポートされている機能のリストから選択して目標に基づいてジョブプランをカスタマイズします。

Important

AWS 適切なアクセス許可がない場合、変換はオペレーションを拒否します。たとえば、寄稿者はメインフレームアプリケーションのジョブ変換をキャンセルしたり、ジョブを削除したりすることはできません。管理者のみがこれらの機能を実行できます。

ジョブを作成するときは、以下の機能から選択できますが、プロジェクトのソースコードの場所であるため、キックオフステップは常に必要です。ワークスペースの最初のジョブ設定では、Amazon S3 バケットへのコネクタを設定する必要があります。

変換の進行状況の追跡

プロセス全体の変換の進行状況は、次の 2 つの方法で追跡できます。

- 作業ログ – AWS 変換が実行するアクションと、人間による入力リクエスト、およびそれらのリクエストに対する応答の詳細なログを提供します。
- ダッシュボード – メインフレームアプリケーション変換の概要を示します。変換されたジョブの数、適用された変換、メインフレームアプリケーションの変換完了までの推定時間に関するメトリクスが表示されます。また、ファイルタイプ別のコード行、各ファイルタイプ別に生成されたドキュメント、分解されたコード、移行計画など、各ステップの詳細を確認することもできます。

コネクタのセットアップ

AWS Transform はコネクタを使用して、メインフレームのモダナイゼーション関数に必要なアカウントのリソースにアクセスします。コネクタは、ワークスペースで最初に行うジョブで自動的に設定されます。選択したジョブプランに応じて、AWS Transform はコネクタを作成するようにガイドします。

メインフレームの再考コネクタ

メインフレーム再設計コネクタは、評価ワークフローと再設計ワークフローを実行するジョブ用です。S3 バケットを使用して変換リソースにアクセスして保存し、Amazon Neptune クラスターを使用して抽出されたアーティファクトを保存します。Neptune クラスターは、ジョブに関する抽出されたすべてのアーティファクトを保存する統合ナレッジグラフとして機能し、ナレッジグラフは AWS 変換チャットインターフェイスのすべての質問に回答します。

AWS Transform が Elastic Network Interface (ENI) を作成できる VPC に Amazon Neptune クラスターをデプロイします。ENIs これらは、AWS Transform が VPC 内の Neptune と安全に通信してデータをロードし、ナレッジグラフをクエリできるようにするネットワーク接続です。

AWS CloudFormation

メインフレームの再設計コネクタに必要なリソースを作成するには、次の CloudFormation テンプレートを使用することをお勧めします。CloudFormation テンプレートの実行時にコネクタに使用する予定のものと同一 S3 バケットを使用します。CloudFormation テンプレートの使用の詳細については、[「テンプレートの使用」](#)を参照してください。

CloudFormation テンプレートをダウンロードするには、[neptune-kg-setup.yaml](#) を選択します。

テンプレートは次のパラメータを受け入れます。

- BucketName – 必須。Neptune バルクローダーアクセスの S3 バケット名。Transform コネクタで使用する予定のものと同じ S3 AWS バケットを使用する必要があります。
- KmsKeyId – オプション。Neptune 暗号化用の KMS キー。キー ID、ARN、エイリアス名、またはエイリアス ARN を受け入れます。省略した場合、デフォルトで AWS マネージドキーになります。

コネクタを設定するときは、次の詳細を指定します。

- S3 バケット ARN
- Neptune クラスター ARN
- Neptune クラスターリソース ID
- アプリケーションサブネット IDs
- アプリケーションセキュリティグループ ID
- Neptune S3 バルクローダーロール ARN

Important

CloudFormation テンプレートの出力タブには、コネクタの設定時に AWS 変換に提供する必要がありますの詳細が含まれています。

カスタム設定

既存の VPC または他のツールを使用して必要なインフラストラクチャを作成する場合は、環境が次の要件を満たしている必要があります。

Neptune クラスター

- エンジンバージョン 1.4.5.1 以降の Amazon Neptune Serverless
 - IAM 認証が有効
 - ストレージ暗号化が有効 (AWS マネージドキーまたはカスタマーマネージド KMS キー)
- サーバーレススケーリング設定 (推奨: 1 ~ 128 NCUs)
- (推奨) 高可用性と読み取りパフォーマンスの向上のために、2 番目のアベイラビリティーゾーンのリードレプリカ

ネットワーク

- DNS サポートと DNS ホスト名が有効になっている VPC
- 別々のアベイラビリティーゾーンに少なくとも 2 つのサブネット。AWS Transform は、これらのサブネットに ENIs を作成して Neptune クラスターにアクセスします。このクラスターは、同じサブネット内または専用サブネット内に存在できます。VPC エンドポイントとアプリケーション ENIs に対応するためにサブネットのサイズを /20 に設定します。
- VPC インターフェイスエンドポイント (AWS PrivateLink) を介したサブネットから次の AWS サービスへのネットワーク接続。
 - AWS Transform Agents API – メインフレームモダライゼーションジョブを調整します
 - Amazon Bedrock ランタイム – 分析と推論のための基盤モデルを呼び出します
 - Amazon Relational Database Service (Amazon RDS) – Neptune クラスター管理
 - Amazon Elastic Compute Cloud (Amazon EC2) – ENIs の管理
 - Amazon CloudWatch – AWS 変換オブザーバビリティのメトリクス
- サブネットに関連付けられたルートテーブルにアタッチされた S3 ゲートウェイエンドポイント。AWS Transform はこのエンドポイントを使用して、S3 バケットから Neptune クラスターにデータをロードします。

セキュリティグループ

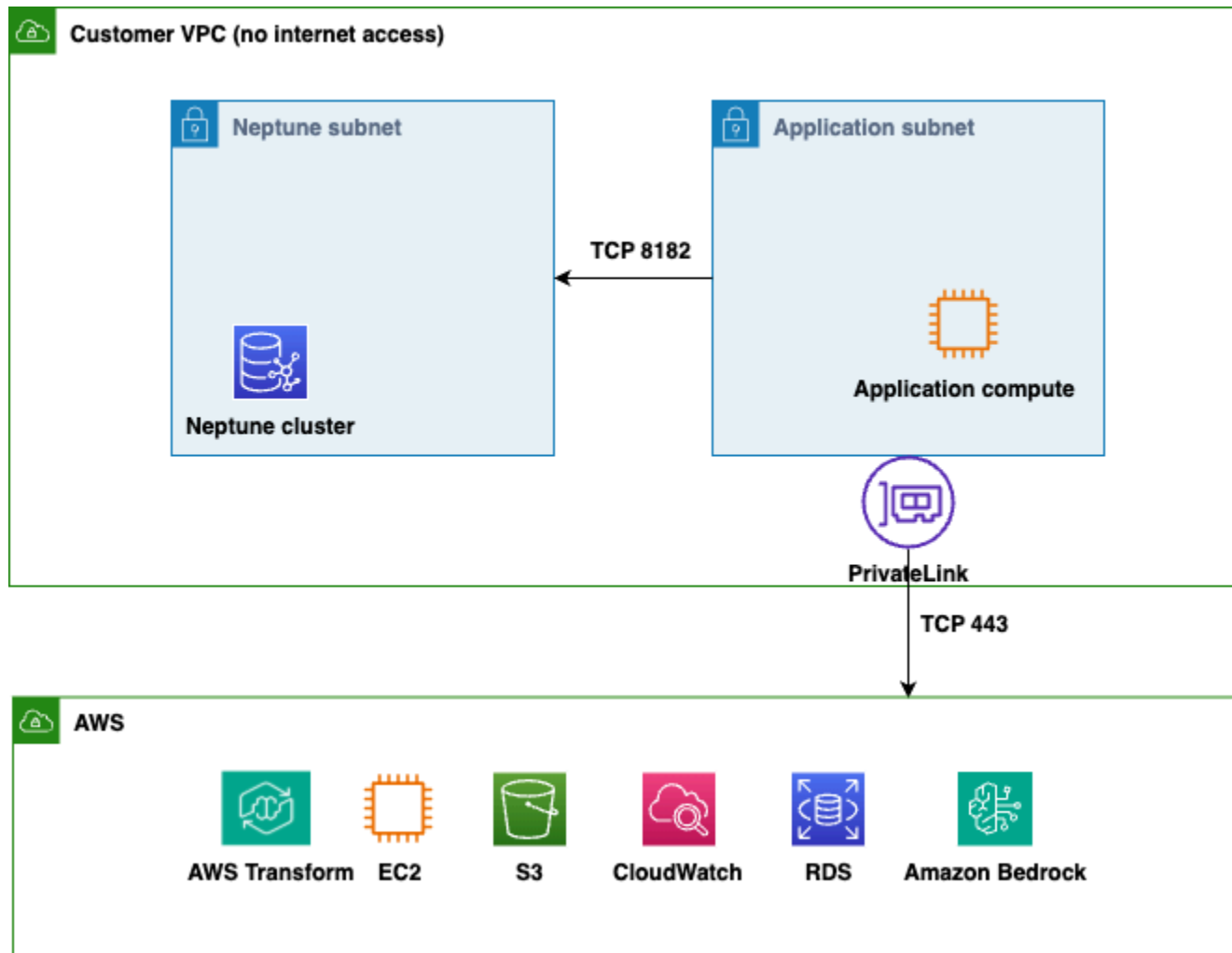
セキュリティグループは、サブネットではなく ENIs にアタッチされます。AWS Transform ENIs と Neptune が同じサブネットに存在する場合でも、別々のセキュリティグループがそれらの間のアクセスを制御します。

- 以下を許可するセキュリティグループ (AWS Transform によって作成された ENIs にアタッチ)。
 - Neptune セキュリティグループへのアウトバウンド TCP 8182
 - VPC エンドポイントに到達するためのアウトバウンド TCP 443
- 以下を許可するセキュリティグループ (Neptune クラスターにアタッチ)。
 - アプリケーションセキュリティグループからのインバウンド TCP 8182
- 以下を許可する VPC エンドポイントセキュリティグループ (インターフェイスエンドポイント ENIs にアタッチ)。
 - アプリケーションセキュリティグループからのインバウンド TCP 443

IAM

- S3 バケットで `s3:GetObject`、`s3:ListBucket`、および `rds.amazonaws.com` する `s3:GetBucketLocation` の信頼ポリシーを持つ IAM ロール
- ナレッジグラフデータを一括ロードするには、このロールを Neptune クラスターに関連付ける必要があります

次の図は、推奨されるアーキテクチャを示しています。



S3 コネクタ

カスタムプランのジョブの場合、AWS Transform は S3 コネクタを使用できます。S3 コネクタは、変換リソースにアクセスして保存する S3 バケットと、インデックス付き出力用の S3 ベクトルバケットを使用します。

⚠ Important

データはワークスペースの AWS 変換アーティファクトストアに保存および保持され、ジョブの実行にのみ使用されます。

S3 ベクトルバケットのオプション設定。

S3 ベクトルバケットが利用可能なリージョンでは、AWS Transform はジョブ出力の検索可能なベクトルエンコードをアカウントのこの S3 ベクトルバケットに保存し、AI を活用した検索とチャットエクスペリエンスを提供します。データはこのジョブの外部では使用されず、モデルのトレーニングにも使用されません。これを有効にするには、S3 ベクトルバケットを作成して提供する必要があります。AWS Transform は、このバケットへの書き込みに必要なアクセス許可を持つロールを自動的に作成してアタッチします。

S3 バケット CORS アクセス許可

コネクタを設定したら、次の CORS ポリシーを S3 バケットに追加して、AWS 変換コンソールでアーティファクトを直接表示および比較できるようにします。このポリシーが正しく設定されていない場合、AWS 変換のインライン表示またはファイル比較機能を使用できなくなる可能性があります。

```
[
  {
    "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
  }
]
```

```
}  
]
```

ジョブプランの評価と再考

評価と再考のジョブプランは、メインフレームアプリケーションをモダナイズするための事前定義されたワークフローです。コードベースを評価してビジネス機能を特定し、選択した機能を再考するという2つのフェーズをガイドします。代わりに個々の機能を選択するには、カスタムジョブプランを使用します。

評価

メインフレームのモダナイゼーションジャーニーは、通常、コンテンツを特定し、関係と依存関係を理解するためのコードベースの評価から始まります。この評価の後、モダナイズするソースコードの一部の分解と選択を開始できます。分解の境界は、ビジネス機能に基づいて決定されます。

ビジネス関数カタログは、コードベースを個別のビジネス関数に分解します。これは、各ビジネス関数を最初から最後までマッピングする決定論的なデータパスの基礎の上に構築されます。データパスは、ビジネストリガーからメインフレームコード内の最終書き込みまで、データが取るルートです。分析を1つのエントリポイントに制限するのではなく、システムはバッチジョブ、CICS トランザクション、およびそれらの共有データストアにまたがって、一貫した業務単位を特定します。

ビジネス機能とは、ビジネストリガーから始まり、測定可能なビジネス成果で終わる一連の作業です。AWS Transform は、事業部門が行動できるビジネス機能のカタログを提供し、最初にモダナイズする内容について情報に基づいた意思決定を可能にします。

ビジネス機能を使用してコードベースを評価してモダナイズするメリット:

- 完全な可視性 – バッチジョブ、オンライントランザクション、データストアがビジネスにとって意味のある機能にどのように接続するかを理解します。
- 実用的な境界 — 特定された各ビジネス機能は、line-of-businessステークホルダーがモダナイゼーションの取り組みについてレビューし、優先順位を付けることができる自己完結型ユニットです。
- コードから仕様までの一貫性 – ビジネス関数を定義するのと同じコードブロックを使用して仕様を生成し、翻訳レイヤーや調整は必要ありません。
- 検出時間の短縮 – 自動検出は、数か月にわたる手動の部族知識インタビューを体系的で反復可能な分析に置き換えます。

チャットを通じて提供された評価の結果

チャットを通じて提供される評価の結果には、データパスの数、バッチトランザクションと CICS トランザクションに関して各関数が何にまたがるか、ビジネスの説明など、ビジネス関数のリストが含まれます。Amazon S3 バケットで生成されたアーティファクトへのリンクが提供され、チャットからアクセスすることもできます。2 つのアーティファクトが提供されます。ビジネス機能の詳細とビジネス機能の概要です。

- ビジネス機能の概要 – ビジネス機能のリストと、コード内の要素が実行している関数の自然言語の説明を提供します。
- ビジネス関数の詳細 – ソースコード間の関係を示すインタラクティブなグラフを提供します。

グラフを操作するには、要素を選択してその詳細を表示するか、要素を開いてその詳細を調べます。各ページには、そのレベルのすべての要素の概要と、以下を含む各コンポーネントを操作して詳細を学ぶためのグラフィカルインターフェイスがあります。

- ビジネス機能 – どのビジネス機能が相互に接続されているかを示し、各ビジネス機能の概要を提供します。ビジネス機能を選択すると、次の情報が提供されます。
 - データパスの数
 - コードの行数
 - ビジネスの説明: コード内の要素が実行する関数を自然言語で記述するビジネス機能の概要。
 - インターフェイス: ビジネス関数内のデータ要素の関係 (交換や書き込みなど) について詳しく説明します。
- ビジネス関数の詳細 – 単一のビジネス関数内のデータパス間の接続を示し、ビジネス関数の構成をより深く掘り下げることができます。要素を選択すると、その要素がビジネス関数全体でどのように接続されるかと、以下の詳細が表示されます。
 - データストア: データストアのライターとリーダー。
 - その他の要素: 自然言語で、データパス内で処理されるアクションと結果を記述する関連データパスの概要、プログラムのリスト、リーダーとライター。
- データパス – 1 つのデータパスの詳細と、データパス内の要素の相互関係を示します。
 - 説明: データパス内で処理されるアクションと結果を自然言語で記述するデータパスの概要。
 - データパスの内容:
 - エントリポイント
 - reads

- writes
- プログラム

チャットを通じて、出力を調べて各ビジネス関数に含まれる要素を理解し、ビジネス関数グラフを通じて境界を調べることができます。

モダナイゼーションの境界を特定すると、チャットは再考するビジネス機能を選択するように求めます。1 つのビジネス機能または複数のビジネス機能を選択できます。AWS Transform は、モダナイゼーションジャーニーを続行するために、選択した境界と必要な評価出力を再考フローに送信します。

最初のモダナイゼーションが完了したら、チャットを使用してモダナイズする追加のビジネス関数のセットを選択できます。

再考

再考フェーズでは、評価によって特定されたビジネス機能を 1 つまたはすべて選択し、モダナイゼーションに備えることができます。これにより、ビジネスロジックの抽出がトリガーされ、選択したビジネス機能の要件が生成されます。生成された要件は、フォワードエンジニアリングの主な入力であり、レガシーシステムの理解を、モダナイズされたアプリケーションが提供する必要があるものの正確な仕様に変換します。

ビジネスロジックを抽出する

1 つ以上のビジネス関数を選択すると、AWS Transform は選択したビジネス関数のビジネスロジックの生成を開始します。抽出が完了すると、AWS Transform はダウンストリーム用に結果を JSON 形式で Amazon S3 バケットに保存します。抽出の進行状況をモニタリングし、AWS 変換コンソール内で直接問題を確認できます。

ビジネスロジックの抽出結果を確認するには

1. 左側のナビゲーションペインで、ビジネスロジックの抽出ステップを選択して展開します。
2. ステップの詳細から以下を表示します。
 - S3 バケットリンク: 抽出されたビジネスロジック結果が JSON 形式で保存される場所。
 - 問題: ビジネスロジック抽出プロセス中に発生した問題のリスト。

抽出中にファイルで問題が発生した場合、影響を受けるファイルごとに次の詳細が表示されます。

- ファイル名: 問題が発生したファイルの名前。
- ファイルタイプ: ファイルのタイプ (COBOL や JCL など)。
- ファイルパス: アプリケーション内のファイルの場所。
- ステータス: ファイルの抽出の現在のステータス。
- 詳細: 発生した問題の説明。

Note

抽出ステップが完了すると、AWS Transform は自動的に要件の生成を開始します。このプロセスを開始する前に、追加の入力は求められません。

要件の生成

ビジネスロジックの抽出が完了すると、Generate Requirements エージェントは抽出されたアーティファクト (コード分析、データ分析、ビジネスルール) を消費し、モダナイゼーション要件を生成します。これらの要件は、テクノロジーに依存せず、テスト可能な許容基準を含む正式な機能仕様です。AWS Transform は、選択したビジネス機能ごとに 1 つの requirements.md ファイルを生成し、出力を Amazon S3 バケットに保存します。出力の Amazon S3 の場所を表示するには、「要件の生成」ステップを選択します。

各 requirements.md ファイルは、次の部分に分かれています。

- タイトル – ビジネス関数とそれが表すワークフローに名前を付けます。
- グローバル前提条件 – 処理前および処理中にワークフロー全体で が当てはまる必要がある条件。設定可能な値はパラメータプレースホルダーとして表示されます。
- 番号付きワークフローセクション – 各セクションは、ビジネス機能の個別のステージを表します。各セクションには以下が含まれています。
 - ロールとしての目標、ロールが実行するアクション、結果を示すユーザーストーリー。
 - EARS (Easy Approach to Requirements Syntax) 形式の要件のリスト。それぞれに一意の識別子があります。

Note

2 番目のビジネス関数セットを選択すると、新しく選択した関数の再考ステップが再開されます。その後、コンソールには新しい選択の結果のみが表示されるため、以前のビジネス

関数のビジネスルールの問題や、ビジネスロジックと要件の出力を保存する Amazon S3 リンクは表示されなくなります。以前の出力は失われません。アーティファクトタブまたは Amazon S3 バケットから引き続きアクセスできます。

変換アーティファクト間のトレーサビリティ

AWS 変換は、評価および再考ワークフロー全体で生成するアーティファクト間のトレーサビリティを維持します。トレーサビリティにより、抽出したビジネスルール、または要件を生成した元のソースコードに直接、各要件を追跡できます。

traceability.yaml ファイルを使用して、各ビジネス関数のすべてのトレーサビリティの詳細を表示できます。各ビジネスルールが要件にどのようにマッピングされるかについて説明します。

インタラクティブなトレーサビリティのために、VS Code および Open VSX 互換エディタで IDE プラグインを使用できます。詳細については、[「デベロッパーツール」](#)を参照してください。

ジョブプランを再考する

モダナイゼーションの対象となり、再考する準備ができているアプリケーションの場合、スタンドアロンの再考ジョブを実行できます。このジョブは、コードとデータの分析、ビジネスロジックの抽出、ビジネスドメインの識別、各ビジネスドメインのモダナイゼーション要件の生成を行います。

カスタムジョブプラン

カスタムジョブプランでは、実行する機能を選択して、独自のモダナイゼーションワークフローを構築できます。これにより、含まれる機能を完全に制御できます。モダナイゼーションの目標に基づいて、次の機能から選択します。一部の機能は、最初に実行される他の機能によって異なります。たとえば、他のほとんどの機能にはコード分析が必要です。

コードを分析する

Amazon S3 バケットパスを AWS Transform と共有すると、ファイル名、ファイルタイプ、コード行、パスなどの詳細を含む各ファイルのコードが分析されます。

Note

分析コードの結果は、ジョブまたはワークスペースレベルのアーティファクトタブからダウンロードできます。ジョブレベルでは、左側のナビゲーションメニューの「Artifacts」オ

プシオンに移動し、「results」フォルダを開くか、ワークスペースレベルでジョブの名前を見つけ、「results」フォルダを開きます。これにより、手動分類ワークフローの分類ファイル、アセットリスト、依存関係 JSON ファイル、欠落しているファイルのリストを含む zip ファイルをダウンロードします。

ジョブプランで、左側のナビゲーションペインでコードの分析を選択して結果を表示します。コード分析の結果は、いくつかの方法で表示できます。

- リストビュー – メインフレームに変換する Amazon S3 バケット内のすべてのファイル
- ファイルタイプビュー – ファイルタイプごとに表示される Amazon S3 バケット内のすべてのファイル。サポートされているファイルタイプのリストについては、[「サポートされているファイル」](#)を参照してください。
- フォルダビュー – フォルダ構造に表示される Amazon S3 バケット内のすべてのファイル。

ファイルの結果内で、AWS Transform は選択したファイルビューに応じて次の情報を提供します。

- 名前
- ファイルタイプ
- 総コード行数
- ファイルパス
- コメント行
- 空の行
- 有効なコード行
- ファイルの数
- 循環的複雑さ - 循環的複雑さは、プログラムのソースコードを通過する直線的に独立したパスの数を表します。AWS 変換では、ファイルごとに循環的複雑さが表示されます。

欠落ファイル – メインフレームモダナイゼーションコード分析の欠落ファイル。これらのファイルは、Amazon S3 バケットのソース入力の一部として追加し、より適切でまとまりのある結果を得るために分析ステップを再実行することをお勧めします。

同一名 – AWS 変換では、同じ名前、場合によっては同じ特性 (コード行数など) を共有するファイルのリストが提供されます。一度に 2 つのファイルの内容の違いを比較することはできません。

重複 IDs – Cobol プログラムでは、プログラム ID フィールドはファイルの一意の識別子として機能します。この ID は、プロジェクト全体でプログラムを呼び出すために使用されるため、一意である必要があります。ただし、名前は異なるがプログラム ID が同じ COBOL ファイルを持つプロジェクトもあります。評価中にこれらのファイルのリストを取得すると、すべてのプログラム間の依存関係を理解するのに役立ちます。

 Note

これは COBOL コードとファイルに固有です。

IDs が重複しているプログラムがある場合は、これらのファイルのプログラム IDs を変更して、COBOL コード内のそれぞれの一意の識別子を持つことをお勧めします。その後、ジョブを再実行して、より正確で包括的なコード分析結果を得ることができます。

重複 IDs を解決することで、次のことが可能になります。

- コードの明確性と保守性の向上
- プログラム呼び出しでの潜在的な競合を減らす
- 依存関係マッピングの精度を高める
- 将来のモダナイゼーションの取り組みを簡素化する

コードベースの問題 – モダナイゼーションプロジェクトを続行する前に解決する必要がある、コードベース内で検出された潜在的な問題。これらの問題には、関連するステートメントを含む参照の欠落や、コード内のサポートされていないリンクが含まれる可能性があります。

分類の更新 – 手動再分類では、新しい分類で JSON ファイルをアップロードすることで、一括更新機能を使用してファイルを再分類できます。

 Important

これは、UNKNOWN および TXT ファイルでのみ使用できます。

再分類後、AWS Transform は次の操作を行います。

1. 分類結果を更新します。

2. 新しいファイルタイプで依存関係分析を再実行します
3. 影響を受けるすべての分析結果を更新します

Note

ファイルは、最初の分析ループが完了した後にのみ再分類できます。

インラインビューワーとファイルの比較

インラインビューワーは、メインフレームの AWS 変換機能の機能であり、2 つの主要な視覚化機能を提供します。

- ファイルビュー: ジョブから選択したレガシーファイルのコンテンツを表示する
- ファイル比較: 2 つのレガシーファイルの side-by-side 比較する

入力ファイルの表示

コードを分析するステップでファイルを表示するには

- 「コード分析結果を表示する」で、リストのチェックボックスを使用してファイルを選択します。

アクションの表示ボタンを選択します (1 つの項目が選択されている場合に有効)。

ファイルコンテンツは、ファイルビューコンポーネントの画面にレンダリングされます。

ファイルの比較

コードを分析するステップでファイルを比較するには

1. 「コード分析結果の表示」で、リスト内のチェックボックスを使用して 2 つのファイルを選択します。
2. 比較アクションボタンを選択します (2 つの項目が選択されている場合にのみ有効)。
3. ファイルはファイル比較コンポーネントに side-by-side 表示されます。

Note

ファイルを比較するために複数のファイルを選択することはできません。

Important

インラインビューまたはファイル比較に問題がある場合は、S3 バケットが正しく設定されていることを確認してください。S3 バケットの CORS ポリシーの詳細については、「」を参照してください [the section called “S3 バケット CORS アクセス許可”](#)。

データ分析

AWS Transform は、データの関係と要素がメインフレームのモダナイゼーションプロジェクトに与える影響を理解するのに役立つデータ分析を提供します。提供される 2 つの出力は次のとおりです。

- データリネージ: データソース、ジョブ、プログラム間の関係をマッピングすることで、データのライフサイクルをトレースします。
- データディクショナリ: レガシーデータ要素の構造メタデータを文書化するリポジトリとして機能します

分析が完了すると、出力ごとに 2 つのタブが存在し、次にデータリネージとディクショナリの両方について以下で説明する複数のビューが表示されます。

Note

データ分析をリクエストすると、AWS Transform はコード分析を実行します。これは、データ分析を実行するために必要です。

データリネージュ

AWS 変換は、モダナイズされるコードベース全体で理解する必要があるデータ関係に基づいて複数のビューを提供します。データリネージュで利用できる 4 つのテーブルビューは次のとおりです。

- データセット: このビューは、読み取り、書き込み、更新、削除の区別に役立つオペレーション追跡など、包括的な影響分析を提供します。
- Db2 テーブル: Db2 テーブルとオペレーションに関連する影響分析を提供します。
- Program-to-data: 各データセットを参照する COBOL プログラムを特定する
- JCL-to-data間の関係: 各データセットを参照する JCL スクリプトを特定する

概要には、データソースの概要、プログラムや JCLs との関係、データソースの活用方法、コードベースで見つかったオペレーションの合計に関する概要情報が記載されています。

データディクショナリ

コードベースに存在するデータソース内のデータ要素を理解することは、さまざまなデータソースがどのように依存しているかを理解する次のステップです。データディクショナリは、正確な変換マッピングのためのビジネス言語の説明を含むフィールドレベルのメタデータを提供するデータカタログです。

- COBOL データ構造: フィールドプロパティやビジネス定義など、コードベースに存在する COBOL コピーブックと INLINE リファレンス全体のフィールド情報を提供します。
- Db2 テーブル: プライマリキーと外部キー、スキーマとインデックスの情報、データ型など、コードベースに存在する Db2 テーブル全体の列とテーブルのプロパティを提供します。

データリネージュとディクショナリの関係は、データソースを選択し、データソースまたはデータディクショナリボタンを使用して、データソースとデータ要素の関係をより深く掘り下げることで利用できます。データリネージュとディクショナリ間のナビゲーションは、「what」-構造と意味、「where」-使用状況と関係を提供するデータリネージュと統合されたデータ可視性を提供します。

アクティビティメトリクス分析

アクティビティメトリクス分析では、タイプ 14、15、30、64、102、110 の System Management Facility (SMF) レコードを分析できます。この分析は、メインフレームアプリケーション内で要素がどのように使用されるかに関するインサイトを提供します。これは、未使用のコードを廃止したり、モダナイズされたアプリケーションのターゲットアーキテクチャに関する決定に役立ちます。ジョブにアクティビティメトリクス分析を含める場合、まずコード分析ステップを完了すると、SMF 出力がより豊富になりますが、必須ではありません。

Note

メインフレームのモダナイゼーションのジョブプランをネゴシエートするときは、まずコード分析と分解のステップを完了すると、より豊富な出力が得られます。

分析用に SMF レコードをオンボードする

SMF 分析の最初のステップとして、Amazon S3 バケット内の SMF レコードの場所を指定する必要があります。最低 13 か月のレコードを指定して、より短い時間枠で見逃す可能性のある年間発生をキャプチャします。13 か月が推奨されますが、検出可能な SMF レコードを持つ時間枠は分析結果を生成します。

SMF 抽出は、次の形式要件を満たしている必要があります。

- タイプ 14、15、30 (サブタイプ 5)、64、102、110 を含めます。
- EBCDIC 形式の raw バイナリファイルを使用します。
- RDW バイトを含めます。

Amazon S3 バケット内の SMF レコードへのリンクを提供し、レコードがソースコードとは別のフォルダにあることを確認します。形式オプションには、圧縮された .zip (最大 600 MB) または .tar.gz (最大 5 GB) が含まれます。SMF レコードが検出されないフォルダへのリンクを指定すると、エラーメッセージが表示され、分析は完了しません。

分析出力

分析出力には、レコードのタイプに応じて、表形式ビューと .csv 出力 (Amazon S3 バケットで使用可能) の 2 つのコンポーネントが含まれます。AWS Transform では、タイプ 30 と 110 の出力が表示されます。Amazon S3 バケットのアーティファクトを通じて、他のタイプのレコードの分析を調べることができます。ユーザーインターフェイスでは、ヘッダーに指定されたレコードの時間範囲が表示され、欠落しているキーの日付を特定できます。例えば、レコードが 5 月 1 日から 10 月 31 日の間にあり、感謝祭後の繁忙期を除外すると、キーレコードがないことを確認できます。ウェブアプリケーションのタイムスタンプには、システムのタイムゾーンと SMF レコードが反映されます。

Amazon S3 で利用可能なアーティファクトは、提供されたレコードで見つかったレコードタイプの分析を提供します。

表形式ビュー

SMF 30 および 110 でのみ使用できる表形式ビューには、バッチジョブと CICS トランザクション用の最大 3 つの主要コンポーネントが含まれています。

- 概要 – キージョブとトランザクションを提供します。
- バッチジョブ/CICS トランザクション分析 – ジョブとトランザクションを集約した分析を提供します。
- コード分析比較 (バッチのみ) – SMF 分析の前にコード分析ステップが実行されたときの比較データを提供します。

検出の概要には、より詳細な分析のために項目をすばやく特定するのに役立つ 3 つのジョブとトランザクションのグループが用意されています。

バッチジョブと CICS トランザクション分析は、ジョブとトランザクションを集約した分析を提供します。分析結果の一部の列はデフォルトで非表示になっています。歯車アイコンを選択すると、追加のフィールドが表示されます。

トランザクションキーは CICS トランザクションの一意のデータ集約を生成し、4 つのフィールドを 1 つのキー値に結合します。

- トランザクション ID
- プログラム名
- SysPlex ID
- SysID

分析出力では、完全なキーまたはキーの任意のコンポーネントで検索できます。

コード分析比較では、SMF レコードまたは分析コードステップのいずれかで見つかったジョブが強調表示されますが、両方には存在しません。これは、SMF レコードの時間枠内に実行されなかったジョブ、またはコード分析ステップに存在しなかったジョブを示します。この出力は、ジョブで SMF 分析の前にコード分析ステップを実行する場合にのみ使用できます。

ベストプラクティス

バッチジョブ (タイプ 30) の包括的な出力を取得するには、JCL ファイル名がジョブ名と一致していることを確認し、コード分析比較で意味のある出力を取得します。

技術ドキュメントの生成

モダナイゼーション中のメインフレームアプリケーションの技術ドキュメントを生成できます。コードを分析することで、AWS Transform は、レガシーシステムに存在するプログラムロジック、フロー、統合、依存関係の説明など、アプリケーションプログラムの詳細なドキュメントを自動的に作成できます。このドキュメント機能は、知識のギャップを埋めるのに役立ちます。これにより、アプリケーションを最新のクラウドアーキテクチャに移行する際に、情報に基づいた意思決定を行うことができます。

Note

技術ドキュメントの生成をリクエストすると、AWS Transform は、ドキュメントの生成に必要なコード依存関係分析を含むコード分析を実行します。

技術ドキュメントを生成するには

1. 左側のナビゲーションペインの [技術ドキュメントの生成] から [ファイルの選択と設定] を選択します。
2. ドキュメントを生成する Amazon S3 バケット内のファイルを選択し、コラボレーションタブで設定を行います。

Note

選択したファイルのエンコードタイプは同じである必要があります (つまり、すべて同じ CCSID - UTF8 または ASCII)。そうしないと、生成された技術ドキュメントに空のフィールドまたはセクションがある可能性があります。

3. ドキュメントの詳細レベルを選択します。
 - 概要 – スコープ内の各ファイルの概要を示します。また、各ファイルの概要を 1 行で提供します。
 - 機能仕様の詳細 – メインフレームアプリケーション変換スコープ内の各ファイルの包括的な詳細を提供します。詳細には、ロジックとフロー、依存関係、入力と出力の処理、さまざまなトランザクションの詳細が含まれます。

 Note

ドキュメントは COBOL ファイルと JCL ファイルに対してのみ生成できます。

4. [続行] をクリックしてください。
5. AWS Transform がドキュメントを生成したら、コンソールの Amazon S3 バケットパスに従ってドキュメントの結果を確認します。このパスでは、結果が生成および保存されます。
6. ドキュメントが生成されたら、AWS Transform Chat を使用して生成されたドキュメントについて質問し、次のステップを決定することもできます。

ユーザー情報をドキュメントに追加する

```
ARTIFACT_ID.zip
### app/
    ### File1.CBL
    ### File2.JCL
    ### subFolder/
    #   # File3.CBL
### glossary.csv
### pdf_config.json
### header-logo.png
### footer-logo.png
# ...
```

オプションのファイルを zip ファイルに追加して、生成されたドキュメントの品質を向上させ、カスタマイズされた PDF カバーページを提供できます。これらには、次のようなものがあります。

- glossary.csv ファイル: S3 バケットの zip ファイルでオプションの用語集を指定してアップロードできます。用語集は CSV 形式です。この用語集は、顧客の語彙に沿った関連する説明を含むドキュメントを作成するのに役立ちます。サンプルglossary.csvファイルは次のようになります。

```
LOL,Laugh out loud
ASAP,As soon as possible
WIP,Work in progress
SWOT,"Strengths, Weaknesses, Opportunities and Threats"
```

- pdf_config.json: このオプションの設定ファイルを使用して、ヘッダー、フッター、ロゴ、カスタマイズされた情報など、会社の形式と標準に沿った PDF ドキュメントを生成できます。サンプルpdf_config.jsonは次のようになります。

```
{
  "header": {
    "text": "Acme Corporation Documentation",
    "logo": "header-logo.png"
  },
  "customSection": {
    "variables": [
      {
        "key": "business Unit",
        "value": "XYZ"
      },
      {
        "key": "application Name",
        "value": "ABC"
      },
      {
        "key": "xxxxxxxxxxx",
        "value": "yyyyyyyyyyyyyy"
      },
      {
        "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
  }
}
```

```
}  
}
```

- ヘッダー:
 - 表紙 PDF ファイルの場合、デフォルトのテキストはプロジェクト名になります。
 - プログラム PDF ファイルごとに、デフォルトのテキストがプログラム名になります。
 - デフォルトのロゴはありません。ヘッダーロゴが設定されていない場合、ロゴは表示されません。
 - フォントサイズとロゴサイズは、単語数またはロゴファイルサイズに基づいて動的に変更されます。
- カスタムセクション:
 - カスタムセクションが設定されていない場合、PDF からは省略されます。
 - リンクはクリック可能である必要があります。
- フッター:
 - フッターにはデフォルトのテキストやロゴはありません。
 - 特に明示的に設定されていない限り、ページ番号はデフォルトでフッターに表示されます。
 - フォントサイズとロゴサイズは、単語数またはロゴファイルサイズに基づいて動的に変更されます。

ドキュメントインラインビューワーを生成する

PDF ファイルは、技術ドキュメントの生成ステップで確認できます。

PDF ファイルを表示するには

1. ドキュメント結果の確認タブに移動します。
2. 生成された PDF のリストで PDFs。
3. ライルを選択し、ファイル名にオーバーレイされたリンク要素を表示または選択します。

PDF が AWS Transform で開き、右上の画面を展開するオプションが表示されます。

Note

AWS Transform では、生成された技術ドキュメントの PDF バージョンの XML をダウンロードすることもできます。

Important

ドキュメントインラインビューヤーに問題がある場合は、S3 バケットが正しく設定されていることを確認してください。S3 バケットの CORS ポリシーの詳細については、「」を参照してください [the section called “S3 バケット CORS アクセス許可”](#)。

ビジネスロジックを抽出する

モダナイゼーション中のメインフレームアプリケーションから重要なビジネスロジックを抽出できます。AWS Transform はコードを自動的に分析し、詳細なプロセスフローやアプリケーションに埋め込まれたビジネスロジックなど、重要なビジネス要素を特定して文書化します。この機能は、モダナイゼーションジャーニーで複数の利害関係者にサービスを提供しています。ビジネスアナリストは抽出されたロジックを活用して、正確なビジネス要件を作成し、現在の実装のギャップや不整合を特定できます。デベロッパーは、メインフレームに関する広範な専門知識がなくても、複雑なレガシーシステム機能をすばやく理解できます。

Note

ビジネスロジックの抽出をリクエストする場合。AWS Transform は、ビジネスロジックの抽出を実行するために必要なコード依存関係やエントリポイント分析を含むコード分析を実行します。

ビジネスロジックを抽出するには

1. 左側のナビゲーションペインのビジネスロジックの抽出で、設定の構成を選択します。
2. コラボレーションタブで、ビジネスロジックを抽出する方法を選択します。
 - アプリケーションレベル: すべてのビジネス機能、トランザクション、バッチジョブ、ファイルのビジネスドキュメントを生成します。これにより、アプリケーション内のすべてのファイルが選択されます。

- ファイルレベル: ファイルテーブルから選択したファイルに対してのみビジネスドキュメントを生成します。

 Note

- どちらのオプションでも、詳細な機能仕様を含めるを選択して、AWS 変換に選択したファイルのコントロールフローと包括的なビジネスルールを含めることができます。
- 選択したファイルのエンコードタイプは同じである必要があります (つまり、すべて同じ CCSID - UTF8 または ASCII)。そうしないと、生成されたドキュメントに空のフィールドまたはセクションがある可能性があります。
- ドキュメントは COBOL ファイルと JCL ファイルに対してのみ生成できます。
- アプリケーションレベルでは、CICS トランザクションとバッチジョブで使用されるプログラムはグループ化され、他のすべてのプログラムは未割り当てとして分類されます。

3. [続行] をクリックしてください。
4. AWS Transform がビジネスロジックを抽出すると、結果を Amazon S3 バケットに JSON 形式で保存し、オンラインで表示できます。

 Note

生成されたビジネスルールファイルの数は、最初の選択数よりも多い場合があります。選択したファイルによっては、ビジネスルールの抽出がトリガーされ、追加の依存ファイルが含まれる場合があります。依存ファイルは結果テーブルにも表示されます。

抽出されたビジネスドキュメントをインラインで表示する

ビジネスロジックは、ビジネスルールの抽出ステップで表示できます。以下の手順に従ってください。

1. ビジネスロジックの抽出結果の確認に移動します。
2. テーブルから表示するドキュメントを選択し、結果の表示ボタンをクリックします。

ビジネスドキュメントページが新しいブラウザタブで開きます。

分解

プログラムとコンポーネント間の依存関係を考慮したドメインにコードを分解できます。これにより、関連するファイルとプログラムを同じドメイン内で適切にグループ化できます。また、分解プロセス中にアプリケーションロジックの整合性を維持するのにも役立ちます。

Note

分解をリクエストすると、AWS Transform は分解を実行するために必要なコード依存関係分析を含むコード分析を実行します。また、より良い結果を得るには、分解前にビジネスロジック抽出を実行することをお勧めします。

アプリケーションの分解を開始するには:

1. 左側のナビゲーションペインからコードを分解を選択します。

Note

未割り当ての 1 つのドメインは、ドメインに関連付けられていないすべてのファイルに対して自動的に作成されます。最初のナビゲーションでは、アプリケーションレベルのビジネスロジック抽出からドメインが提案された場合を除き、すべてのファイルを未割り当てに関連付ける必要があります。

2. アクションメニューから新しいドメインを作成し、ドメインの作成を選択します。
3. 「ドメインの作成」で、ドメイン名とオプションの説明を入力し、一部のファイルをシードとしてマークします。
 - CICS 設定ファイル (CSD) とスケジューラ設定ファイル (SCL) は、自動シード検出に使用できません。
 - 1 つのドメインを共通のコンポーネントとしてのみ設定することもできます。このドメインのファイルは、複数のドメインに共通です。
4. [作成] を選択します。

Note

シードとして異なるファイルを持つ複数のドメインを作成できます。

5. すべてのドメインとシードを確認したら、分解を選択します。
6. AWS 変換はソースコードファイルをチェックし、同様のユースケースと高いプログラミング依存関係を持つプログラムとデータセットを使用してドメインに分解します。

AWS 変換では、分解されたドメインの表形式およびグラフビューを依存関係として使用できます。グラフビューには 3 つのオプションがあります。

- ドメインビュー – さまざまなドメインが互いにどのように関連しているかを視覚的に表示できます。
- 依存関係ビュー – 各ドメインのすべてのファイルを複雑な依存関係グラフとして表示できます。ドメインに追加されたノードが同じドメインのシードから情報を受信しなかった場合、このノードは未割り当て (ノードは情報を受信しなかった)、切断 (シード情報を受信しなかったサブグラフ内)、または別のドメイン (ノードは少なくともそのドメインから情報を受信) のいずれかに予測されます。
- サブグラフビュー – ユーザーはサブグラフを作成してノードのサブセットを視覚化し、ノードのコレクションのリレーショナルな影響と境界をより明確に理解できます。
 - サブグラフを作成するには、ノードのグループを選択し、ツールバーからサブグラフの抽出オプションを選択します。
 - マージするサブグラフは、マージする 2 つのサブグラフに加えて、新しい 3 番目のサブグラフを作成するときに使用できます。

Note

現在のドメイン構造が気に入らない場合は、これらのステップを繰り返してドメインを追加したり、作成済みのドメインを別のシードセットで再設定したりできます。

7. 完了したら、続行を選択します。

シード

シードは、分解コードフェーズの基本的な入力です。各コンポーネントまたはファイル (JCL、COBOL、Db2 テーブル、CSD、スケジューラファイルなど) は、1 つのドメインにのみシードとして割り当てることができるため、分解プロセス中に明確な境界とアライメントを確保できます。

シードの識別は、アプリケーションまたはポートフォリオの構造によって異なります。一般的なメインフレームのレガシーアプリケーションの場合、シードは、確立された命名規則、スケジューラでのバッチレベルのグループ化、および CICS システムで定義されたトランザクションレベルのグループ

化に従って決定できます。さらに、データベーステーブルはシードとして機能し、分解用の別の構造レイヤーを提供することもできます。

依存関係ファイルのインポートおよび/または更新

分解中に、Transform が実行する依存関係分析によって生成された既存のファイルを置き換える依存関係の AWS JSON ファイルをアップロードできます。

依存関係のエクスポート関数を使用すると、分解ステップで生成された依存関係 JSON ファイルをダウンロードすることができます。ダウンロード後、要件に応じてファイルを変更できます。その後、AWS 変換のアップロード機能を使用して依存関係をインポートできます。これにより、依存関係分析によって生成されたファイルを置き換える依存関係の新しい JSON ファイルをアップロードできます。その後、分解ステップのグラフが更新されます。

依存関係をエクスポート、変更、インポートするには

1. 分解結果の表示ページで、アクションを選択します。
2. ドロップダウンリストで、「その他のアクション」の「依存関係ファイルの更新」オプションを選択します。
3. 依存関係の更新ファイルモーダルで、
 - a. 依存関係ファイルをダウンロード 既存の分析結果から作成された AWS 変換。
 - b. ダウンロードしたファイルで、達成したい内容に基づいて依存関係を変更します。
 - c. 変更後、依存関係ファイルのアップロードボタンを使用して、このファイルを保存してアップロードします。

Note

使用できるファイル形式は JSON ファイルのみです。

4. 次に、インポートを選択します。

AWS Transform は依存関係ファイルをインポートし、入力に基づいて新しい依存関係グラフを作成します。

ドメインのインポートおよび/または更新

分解ステップの前にドメイン、シード、および/またはファイル関係がマッピングされているお客様は、アクションメニューで利用可能なドメインのインポートファイル関数を使用してこのドメイン定義をアップロードできます。この関数がいつ使用されるかの例を以下に示します。

- 別のジョブからの分解を早送りする
- このマッピングを提供する対象分野のエキスパート

ドメインファイルがインポートされると、ユーザーはドメイン定義に対して分解を実行するか、問題がなければドメイン定義を保存して送信できます。

Parent/child/neighborファイル

依存関係グラフでは、プログラムはさまざまなタイプの接続を通じて相互に関連しています。これらの関係を理解することで、メインフレームアプリケーションの変換中にプログラムの依存関係を分析することができます。また、ドメインの境界を理解するのに役立ちます。たとえば、ドメインを選択し、親レベルを1つ選択すると、接続されたノードが表示されます。

親関係 – 親ファイルが他のプログラムを呼び出したり制御したりします。親は、階層内の依存プログラムの上に座っています。親は、1つのレベルまたはすべてのレベルで選択できます。

子関係 – 子ファイルは親プログラムによって呼び出されるか、制御されます。子はファイル階層の親の下に座ります。

近傍関係 – 近傍は同じ階層レベルのファイルです。同じ親プログラムを共有し、相互に直接やり取りする場合があります。

リフォージコード

Reforge は、大規模言語モデル (LLMs) を使用してリファクタリングされたコードの品質を向上させます。最初の COBOL-to-Java 変換は、COBOL の影響を受けたデータ構造と変数名をレガシーシステムから保持しながら、機能的同等性を維持します。Reforge は、最新の Java プラクティスとイディオムに従うようにこのコードを再構築し、COBOL スタイルのコンストラクトをネイティブ Java コレクションと命名規則に置き換えます。これにより、Java 開発者はコードをより読みやすく維持できます。

Note

リフォージのクォータは次のとおりです。

- ジョブあたり 3,000,000 行のコード
- ユーザーあたり 1 か月あたり 50,000,000 行のコード

以下の手順に従って、リファクタリング後にコードをリフォージします。

1. 左側のナビゲーションペインで Reforge java code を選択し、Configure code reforge を選択します。
2. 圧縮されたビルド可能なソースプロジェクトに S3 の場所を指定し、続行を選択します。次の zip 構造を使用します。

```
input.zip
  ### PROJECT-pom
  ### PROJECT-entities
  ### PROJECT-service
  ### PROJECT-tools
  ### PROJECT-web (optional)
  ### pom.xml
```

AWS Transform は zip パッケージを分析して PROJECT-service ディレクトリ内のファイルを検索し、再編成できるクラスの選択可能なリストを提供できるようにします。これらのクラスにはサフィックスがあります ProcessImpl.java。

3. 再編成するクラスの選択ページを完了し、続行を選択します。Worklog タブで再フォージステータスを追跡します。
4. 完了したリフォージの結果を「結果の表示」ページで表示します。これにより、クラスごとのリフォージステータスが表示されます。また、S3 バケット内のリフォージ結果の場所も指定します。

AWS Transform がこの入力を取得すると、再編成の結果を含むダウンロード可能なファイルが提供されます。

これは、リフォージが成功した結果の zip 構造です。

```
reforge.zip
  ### maven_project
  ### reforge.log
  ###tokenizer_map.json
```

- `maven_project` には、再編成されたソースコードが含まれています。
- リファクタリングされたがコンパイルが正常に確定されなかったファイルは、`src/main/resources/reforge/originalClassName.java.incomplete`、という名前が付けられます `originalClassName.java.incomplete`。これらをファイルの元のバージョンと比較して、保存する再編成された関数を選択します。
- 正常にリファクタリングされた AWS 変換に提供されるソースファイルは `src/main/resources/reforge/originalClassName.java.original` され、という名前が付けられます `originalClassName.java.original`。リファクタリングされたバージョンのファイルは、AWS 変換に提供されたソースファイルを置き換えます。

Note

`originalClassName.java` ファイルは、再編成プロセスが成功した場合にのみ、再編成されたファイルに置き換えられます。それ以外の場合は、元のコンテンツが保持されます。

- `reforge.log` には、ジョブの失敗を診断したり、問題が発生した場合 AWS にサポートするために提供したりするために使用できるログが含まれています。
- `tokenizer_map.json` には、ファイルパスやクラス/メソッド名などのデータへのトークン IDs のマッピングが含まれており、プライバシー保護のためにログでトークン化されます。このファイルは、問題が発生した場合に AWS をサポートするために提供できます。

モダナイズされたアプリケーションのテストを計画する

抽出されたコード属性、ジョブの複雑さ、スケジューラパスに基づいて、メインフレームをモダナイズしたアプリケーションのテストプランを作成および管理できます。AWS Transform は、テストするジョブの優先順位を決定し、各テストケースに必要な特定のアーティファクトを特定するのに役立ちます。テスト計画プロセスは、設定、範囲設定、レビューの 3 つの主要なフェーズに分かれています。

テストプランを作成するには

1. テストプランの設定を構成する

- a. 左側のナビゲーションペインの「テストを計画する」で、「設定を構成する」を選択します。

- b. (オプション) ビジネスロジック抽出 (BLE) の S3 パスを指定します。これらのアーティファクトは、テストプランの品質を向上させます。BLE アーティファクトがないと、テストプランの一部のフィールドが不完全のままになる可能性があります。

2. テストプランの範囲を定義する

- a. テストプランに含めるバッチジョブなどのエントリポイントを選択します。
- b. 複数の属性に基づいてジョブをフィルタリングおよびソートします。
 - ビジネス関数 (BRE BLE から抽出)
 - ドメイン (分解フェーズから)
 - ファイルパスと場所
 - カスタム検索条件
- c. テストする個々のジョブまたはグループ全体を選択します。
- d. ジョブの関係と依存関係を確認します。

3. テストプランを確認して調整する

生成されたテストプランは、次のような包括的な情報を提供します。

- 依存関係に基づく優先実行順序
- スケジューラからのジョブグループの割り当て
- テストケースの集計スコアである複雑さスコア
- ビジネスドメインの関連付け
- 循環的な複雑さのメトリクス
- データセットとテーブルの依存関係
- コードメトリクスの行
- ビジネス関数マッピング

テストプランのカスタマイズオプション

テストプランは、特定のニーズに合わせてカスタマイズできます。例えば、以下のことが可能です:

- 複数のエントリポイントを選択して新しいテストケースを作成する
- 既存のテストケースをマージして関連機能を組み合わせる
- より詳細なテストのためにテストケースを分割する

- 不要なテストケースを削除する
- 既存のテストケースのエントリポイントを追加または削除する
- テストケースの説明と属性を変更する
- 実行順序を調整する

詳細なテストケース情報

各テストケースには、その内容と関連するデータセットまたはデータファイルの詳細が表示されます。

- テストスコープの包括的な説明
- 含まれるエントリポイントの完全なリスト
- 複雑さとサイズを示すメトリクスを集約する
- ビジネスルールと自動テストケースガイダンス
- 方向 (入力/出力) によるデータセットとテーブルの依存関係
- インタラクティブな依存関係グラフの視覚化
- 実行の前提条件と要件

データ管理機能

これらの詳細は、テストケースに関連するデータを理解するのに役立ちます。以下の操作を実行できます。

- 入出力方向でデータセットをフィルタリングする
- テスト実行に必要なアーティファクトを特定する
- テストケース間のデータの依存関係を追跡する
- テストプラン全体でデータセットの使用状況をモニタリングする

Note

AWS Transform は、スケジューラの依存関係を自動的に分析し、複雑さスコアを割り当てて、テスト作業の優先順位付けを支援します。複雑さスコアが高いほど、テストプロセス中により徹底的なテストまたは分離が必要なジョブを示します。

ビジネスルールとテストケースガイダンス

AWS 変換テストプランは、ビジネスルールの抽出に基づいてテストケースプランの推奨事項を提供します。

- LLM を使用したビジネスルールの自動処理
- 合成テストケースの生成
- 特定のビジネスシナリオのテストガイダンス
- ルールとテストケース間のトレーサビリティ

最終的なテストプランは、指定された S3 の場所に保存され、テスト戦略を効果的に実行するために必要なすべての情報が含まれています。テストプランをエクスポートして、他のテストツールやドキュメントシステムと統合できます。

Note

合成テストケースガイダンスが提供されますが、実際のテストアーティファクトはガイダンスに基づいて個別に作成する必要があります。テストプランは、テスト戦略の包括的な設計図として機能しますが、テストデータや実行スクリプトは生成されません。

テストケース作成ルール - 概要

一般的なルール

- テストケースには、スケジュール実行順序で 1 ~ 多くの JCLs が含まれます。
- テストケースは、有効なサポートされているスケジューラ (CA7 および Control-M) から作成されます。
- テストは、スケジューラ自体やスケジュールされたタスクではなく、JCLs を実行します。
- スケジュールされたタスクは一意であり、1 つの JCL のみを実行します。
- 1 つの JCL を複数のスケジュールされたタスクで実行できます
- スケジュールになくても 1 つの JCL を存在できます

デフォルトのテストケース作成ルール

- 1 つの JCL がテストケースにある場合:

- JCL がスケジュールに關与していない
- JCL は、複数のブランチに分散するスケジュールされたタスクによって実行されます。
- スケジュール内のブランチには、この JCL のみが含まれます。
- テストケース内の複数の JCL の場合: スケジュールブランチ内の線形実行パスシーケンスを表します。
- JCL がさまざまなスケジュールされたタスクにある場合: JCL は独自の個別のテストケースになります
- JCL が収束スケジュールされたタスク上にある場合: JCL は新しいテストケースを開始できますが、以前のブランチには含まれません
- 欠落JCLs はスキップされ、実行は続行されます (欠落している JCL ポイントでテストケースを切り取るための今後の拡張が予定されています)。

ユーザーオペレーションルール

- テストケースの作成: ユーザーが利用可能な JCLs
 - JCLsスケジュール実行ブランチシーケンスに存在する場合に成功します
 - スケジュールの実行順序に従います
- テストケースに JCL を追加する: ユーザーは利用可能な JCLs から選択します
 - JCL がテストケースのスケジュール実行ブランチ/パスに存在する可能性がある場合に成功します
- テストケースから JCL を削除: ユーザーはテストケースから任意の JCL を削除できます
 - 実行パスのギャップが発生する場合でも許可されます
- テストケースのマージ: ユーザーは結合する 2 つのテストケースを選択します
 - JCLs が同じスケジュール実行ブランチと一緒に存在できる場合に成功します
 - スケジュール実行順序を維持します
- テストケースの分割: ユーザーが分割するテストケースで 1 つの JCL を選択する
 - 分割ポイントフォワードから新しいテストケースを作成します
 - 元のテストケースを変更して、分割ポイントを超えた JCLs を除外
- テストケースの削除: ユーザーは作成されたテストケースを削除できます

Note

異なるスケジューラブランチ/パスからテストケースを作成、追加、またはマージすることはできません。今後の機能強化は、スケジューラの分散/収束タスク以外のオペレーションを許可するように計画されています。

テストデータ収集スクリプトを生成する

JCL スクリプトを生成して、前のステップで作成したテストプランに基づいてメインフレームシステムからテストデータを収集できます。AWS Transform は、包括的なテストに必要なデータセット、データベーステーブル、シーケンシャルファイルのデータ収集スクリプトを自動的に作成します。データ収集プロセスは、入力設定、テストケースの選択、スクリプト設定、スクリプト生成の 4 つの主要なフェーズに分かれています。

テストデータ収集スクリプトを生成するには

1. テストプランの入力を提供する
 - a. 左側のナビゲーションペインの「データ収集をテストする」で、「テストプラン入力を提供する」を選択します。
 - b. [モダナイズされたアプリケーション](#)のテストを計画から、テストプラン JSON ファイルへの S3 パスを指定します。
 - c. 前のジョブステップでテストプランが生成された場合、入力フィールドは事前に入力されます。
 - d. 適切な S3 の場所を指定して、他のジョブからテストプランを選択することもできます。
2. データ収集のテストケースを選択する
 - a. テストプランのテストケースの完全なリストを確認します。
 - b. 複数の属性に基づいてテストケースをフィルタリングおよびソートします。
 - ビジネス関数とドメイン
 - データベーステーブルの依存関係
 - データセットの要件
 - 複雑さメトリクス
 - カスタム検索条件
 - c. 個々のテストケースを選択するか、一括選択オプションを使用します。

- d. エントリポイント、メトリクス、ビジネスルールなどのテストケースの詳細を確認するには、1 つのテストケースをクリックして詳細を確認します。
3. データ収集スクリプトを設定する
 - a. 参照用にサンプルテンプレートと設定ファイルをダウンロードします。
 - AWS Transform は、Db2 データベースアンロード、VSAM ファイル REPRO、およびシーケンシャルデータセット処理のサンプルテンプレートを提供し、プロセスで予想されるテンプレートの種類に関するガイダンスとして使用します。
 - 標準はサイトごとに異なる場合があるため、お客様は独自の標準に準拠したテンプレートを変更または置き換えることが期待されます。
 - これらの変更されたテンプレートは、テストデータ収集で処理できる S3 バケットにアップロードする必要があります。
 - b. 以下を含む変数設定ファイル (JSON 形式) を指定します。
 - ユーザープレフィックスと環境固有の定数
 - データベース設定パラメータ
 - 送信先エンドポイントの設定とデータ転送パラメータ
 - JCL テンプレートで使用するユーザーとで定義されるその他の必須パラメータ
 - c. さまざまなデータ収集方法の JCL テンプレートをアップロードします。
 - Db2 テンプレート: データベーステーブルのアンロード用 (BMC、IBM DSN、またはその他のアンロードユーティリティ用にカスタマイズ)
 - VSAM テンプレート: VSAM ファイル処理用 (通常は REPRO ユーティリティを使用)
 - シーケンシャルデータセットテンプレート: シーケンシャルデータセット、パーティション化されたデータセット、GDGsなどの処理用。

4. 生成されたスクリプトの確認と管理

生成されたスクリプトは、次のような包括的なデータ収集機能を提供します。

- 「前」と「後」のテスト実行データ収集用の個別のスクリプト
- テストケースとデータ型別に整理されたスクリプト構造
- スクリプトは自動的に S3 バケットに保存され、アクセスと転送が容易になります。
- 生成された JCL スクリプトはメインフレーム実行の準備ができています
- 変数の置換は、テンプレートで定義されたユーザー設定に基づいています。

スクリプト生成機能

生成されたスクリプトは、テンプレートと設定に基づいて自動的にカスタマイズされます。

- テンプレートベースの生成: 提供された JCL テンプレートを変数置換で使します
- 環境: 特定のメインフレーム設定を組み込みます
- データ型処理: シーケンシャルデータセット、VSAM ファイル、データベーステーブルに適したスクリプトを作成します。
- テストケースの収集: 「前」と「後」の両方のデータ収集スクリプトを生成します
- シーケンシャルデータセット処理: AWS が提供するサンプルは、ファイル転送機能を提供しますが、これは、サイトで利用可能な圧縮ユーティリティ、または Connect Direct やマネージドファイル転送などのユーティリティにカスタマイズできます。

データ収集戦略

生成されたスクリプトは、包括的なデータ収集戦略をサポートします。

- 連続データセットコレクション: VSAM およびフラットファイル用の REPRO ユーティリティとコピーユーティリティ
- データベーステーブルのアンロード: カスタマイズ可能な Db2 アンロードプロセス
- シーケンシャルデータセット処理: 圧縮、マネージドファイル転送サービスなどのシーケンシャルデータセットのカスタマイズ可能な後処理。
- 依存関係管理: テストケース定義に基づく調整されたコレクション

Note

AWS Transform は、テンプレートと設定に基づいてスクリプトを生成します。メインフレーム環境でを実行する前に、生成されたすべての JCL を確認して、特定のシステム設定とセキュリティ要件との互換性を確認します。

テンプレートのカスタマイズとベストプラクティス

ATX テストデータ収集は、柔軟なテンプレートカスタマイズ機能を提供します。

- 複数ユーティリティのサポート: さまざまなメインフレームユーティリティ (BMC、IBM、DSN) のテンプレートを適応させる

- 変数駆動型設定: 環境固有のパラメータに定数を使用する
- 再利用可能なテンプレート: 一貫したスクリプト生成のための標準化されたテンプレートを作成する
- データ処理: 組織固有のデータ処理要件を組み込む
- セキュリティ統合: 適切なセキュリティとアクセスコントロールを含める
- パフォーマンスの最適化: 効率的なデータ収集と転送のために を設定する

生成された出力構造

生成されたスクリプトは、次の構造で S3 バケットに整理されます。

- テストケースの整理: 関連するテストケース別にグループ化されたスクリプト
- 収集タイミング: 「前」と「後」のデータ収集用にフォルダを区切る
- データ型分類: シーケンシャルデータセット、データベーステーブル、転送によって整理されたスクリプト
- メタデータファイル: 概要情報と実行ガイダンス
- Ready-to-transfer 形式: メインフレームの直接デプロイ用に JCL 形式

最終的なスクリプトコレクションは、指定された S3 の場所に保存され、データ収集戦略を効果的に実行するために必要なすべての JCL が含まれます。スクリプトをダウンロードしてメインフレーム環境に転送したり、自動デプロイプロセスと統合したりできます。

Note

包括的な JCL スクリプトが生成されますが、実際の実行はメインフレーム環境で実行する必要があります。スクリプトはready-to-useデータ収集ツールとして機能しますが、適切なメインフレームアクセスと実行のアクセス許可が必要です。

自動化スクリプトの生成をテストする

テスト自動化スクリプトを生成して、前のステップで作成したテストプランに基づいて、モダナイズされたアプリケーションでテストケースを実行できます。AWS Transform は、テストデータ収集プロセスから収集されたデータを利用する包括的なテストスクリプトを自動的に作成します。テスト自動化スクリプト生成プロセスは、入力設定、テストケースの選択、スクリプト生成結果の 3 つの主要なフェーズで構成されます。

テストスクリプトを生成するには

1. テストプランの入力を提供する

- a. 左側のナビゲーションペインのテスト自動化スクリプト生成で、テストプラン入力の提供を選択します。
- b. [モダナイズされたアプリケーションの](#)テストを計画から、テストプラン JSON ファイルへの S3 パスを指定します。
- c. 前のジョブステップでテストプランが生成された場合、入力フィールドは事前に入力されます。
- d. 適切な S3 の場所を指定して、他のジョブからテストプランを選択することもできます。
- e. システムは、このテストプランをオートメーションスクリプトを生成するための基盤として使用します。

2. スクリプト生成のテストケースを選択する

- a. テストプランのテストケースの完全なリストを確認します。
- b. 複数の属性に基づいてテストケースをフィルタリングおよびソートします。
 - ビジネス関数とドメイン
 - データベーステーブルの依存関係
 - データセットの要件
 - 複雑さメトリクス
 - カスタム検索条件
- c. 個々のテストケースを選択するか、すべてチェックボタンとすべてチェック解除ボタンで一括選択オプションを使用します。
- d. 個々のテストケースをクリックして、エントリポイント、メトリクス、ビジネスルールなどのテストケースの詳細を確認します。

選択したテストケースには、モダナイズされたアプリケーションで実行するための自動化スクリプトが生成されます。

3. 生成されたテスト自動化スクリプトを確認および管理します。

- スクリプト生成の完了を確認する成功メッセージが表示されます。
- 生成されたテストスクリプトは、指定された S3 バケットの場所に自動的に保存されます。

- 生成されたテストスクリプトの完全なリストに、それぞれの S3 ロケーションでアクセスします。
- 各テストケースには、対応する自動化スクリプトが個々の S3 ロケーションに保存されます。
- スクリプトは、モダナイズされたアプリケーション環境でデプロイおよび実行できます。

自動化スクリプト機能のテスト

生成された自動化スクリプトは、包括的なテスト機能を提供します。

- モダナイズされたアプリケーションテスト: スクリプトは、変換されたアプリケーションでテストケースを実行するように特別に設計されています。
- データ統合: 前のテストデータ収集ステップで収集されたテストデータを使用します。これらのデータは、テストケースごとにフォルダにコピーする必要があります。
- 自動実行: スクリプトを使用して、データシンクの設定、テストケースの実行、結果の比較を行うことができます。一部のパラメータはデプロイ環境に応じて設定する必要があります。
- 組織構造: スクリプトは S3 バケット内のテストケースごとに体系的に整理されます。
- Ready-to-deploy 形式: スクリプトはテスト環境に直接デプロイできるようにフォーマットされています

生成された出力構造

生成されたテストオートメーションスクリプトは、次の構造で S3 バケットに整理されます。

- テストケースの整理: 各テストケースには、個々の S3 フォルダに保存されている専用スクリプトがあります。
- 実行対応形式: スクリプトは、環境に応じて変数を設定した後、すぐにデプロイおよび実行できるようにフォーマットされます。
- 一元化されたアクセス: すべてのスクリプトに単一の S3 バケットの場所からアクセスでき、管理が容易

テスト実行戦略

生成されたスクリプトは、包括的なテスト実行ワークフローをサポートします。

- 環境設定: スクリプトには、テストを実行するための初期データを設定する機能が含まれています。

- データ準備: テストケースデータ収集ステップから収集されたテストデータとの統合
- テストケースの実行: モダナイズされたアプリケーションでの個々のテストケースの自動実行
- 結果の比較: テスト結果を比較し、アプリケーションの動作を検証するための組み込み機能

Note

AWS Transform は、テストプランと選択したテストケースに基づいてテスト自動化スクリプトを生成します。スクリプトは、モダナイズされたアプリケーション環境で実行され、前のステップで収集されたテストデータを利用するように設計されています。デプロイ前に生成されたすべてのスクリプトを確認して、特定のアプリケーション設定およびテスト要件との互換性を確認します。

テスト自動化のベストプラクティス

- 環境の検証: スクリプトを実行する前に、モダナイズされたアプリケーション環境が正しく設定されていることを確認します。
- データ検証: 収集フェーズから必要なテストデータが使用可能でアクセス可能であることを検証します。
- スクリプトのカスタマイズ: 特定のテスト要件に応じて、生成されたスクリプトを確認してカスタマイズします。
- 実行モニタリング: テストスクリプトの実行中に適切なモニタリングとログ記録を実装する
- 結果分析: テスト結果を分析し、アプリケーションの問題を特定するプロセスを確立する

テスト自動化スクリプトの最終コレクションは、モダナイズされたアプリケーション機能を検証するための完全なテストフレームワークを提供します。スクリプトは、継続的なテストプロセスに統合することも、アプリケーション検証ワークフローの一部として実行することもできます。

AWS Transform のデプロイ機能

AWS Transform は、ready-to-use Infrastructure as Code (IaC) テンプレートを提供することで、モダナイズされたメインフレームアプリケーションのクラウド環境をセットアップするのに役立ちます。AWS Transform チャットインターフェイスを使用すると、コンピューティングリソース、データベース、ストレージ、セキュリティコントロールなどの重要なコンポーネントを作成する構築済みのテンプレートにアクセスできます。テンプレートは、CloudFormation (CFN) AWS Cloud

Development Kit (AWS CDK)、Terraform などの一般的な形式で利用できるため、インフラストラクチャを柔軟にデプロイできます。

これらのテンプレートは、モダナイズされたメインフレームアプリケーションの環境を設定するために必要な時間と専門知識を削減する構成要素として機能します。これらのテンプレートはニーズに合わせてカスタマイズでき、デプロイ環境を構築するための基盤となります。

IaC テンプレートを取得するには、AWS Transform チャットの「Infrastructure-as-Code」テンプレートに、希望するモダナイゼーションパターン (メインフレームリファクタリングの AWS 変換など)、希望するトポロジー (スタンドアロンか高可用性か)、希望する形式 (CloudFormation か Cloud Development Kit か Terraform か) を明確にするよう依頼します。

リファクタリング後にモダナイズされたアプリケーションを構築してデプロイする

AWS Transform でリファクタリングプロセスを完了したら、モダナイズされた Java アプリケーションを構築してデプロイできます。このガイドでは、モダナイズされたコードの取得、環境の設定、アプリケーションのデプロイとテストについて説明します。

Note

ここで提供されるガイダンスに加えて、AWS 変換生成コードパッケージには、[開発者ランタイム](#)で IDE (統合開発環境) を設定する手順を提供する AWS 自動リファクタリング開発環境のセットアップドキュメントが含まれます。

トピック

- [前提条件](#)
- [ステップ 1: モダナイズされたコードを取得する](#)
- [ステップ 2: モダナイズされたアプリケーションを構築する](#)
- [ステップ 3: テスト環境を設定する](#)
- [ステップ 4: モダナイズされたアプリケーションをデプロイする](#)
- [ステップ 5: モダナイズされたアプリケーションをテストする](#)
- [その他の例](#)

前提条件

開始する前に、以下の準備が整っていることを確認します。

- Transform を使用してリファクタリングジョブを正常に完了 AWS しました。
- モダナイズされたコードへのアクセス。
- [Apache Maven](#) や [Apache Tomcat](#) など、開発マシンにビルドソフトウェアツールスタックをインストールして設定します。ランタイムバージョンの詳細については、[AWS 「メインフレームのランタイムリリースノートの変換」](#) を参照してください。
- Amazon Corretto または Java ランタイムのバージョンをインストールして設定します。Amazon Corretto のインストールの詳細については、[「Amazon Corretto 24」](#) を参照してください。
- 必要に応じて、ランタイムコンポーネントの Amazon Aurora PostgreSQL データベースを作成および設定するためのアクセス。Aurora PostgreSQL データベースの作成の詳細については、[「Amazon Aurora PostgreSQL の使用」](#) を参照してください。
- ランタイム環境にアプリケーションをデプロイするための管理アクセス。
- AWS 自動リファクタリングソリューションでモダナイズされたアプリケーションの基本的な概念について、[AWS メインフレームランタイムの変換](#) の概念を確認しました。

ステップ 1: モダナイズされたコードを取得する

モダナイズされたコードを取得するには

1. 生成されたコードパッケージをダウンロードして抽出します。
2. を開き codebase/app-pom/pom.xml、必要なランタイムエンジンバージョンを書き留めます。例: `<gapwalk.version>4.6.0</gapwalk.version>`。
3. ダウンロードしたコードパッケージから AWS 自動リファクタリング開発環境の設定ドキュメントを参照してみます。

ステップ 2: モダナイズされたアプリケーションを構築する

モダナイズされたアプリケーションを構築するには

1. メインフレームの AWS 変換ツールボックスからランタイムにアクセスします。
2. 適切なランタイムバージョン (で識別 [the section called “ステップ 1: モダナイズされたコードを取得する”](#)) をダウンロードしてローカル開発マシンにインストールします。

Note

ランタイム依存関係をローカルマシンにインストールするための追加情報は、AWS「[自動リファクタリング開発環境のセットアップ](#)」ドキュメントのセクション 3.1 で入手できます。

3. コマンドプロンプトを開き、アプリケーションのルートディレクトリに移動します。
4. モダナイズされたアプリケーションのデプロイ可能なパッケージを構築するには、Maven build コマンドを実行します。

```
mvn package
```

モダナイズされたコードの基本的な構成の詳細については、[アプリケーション組織](#)のページを参照してください。たとえば、フロントエンドウェブアプリケーションを含むモダナイズされたアプリケーションの場合、ランタイムコンポーネントに加えて、少なくとも次のデプロイ可能な .war アグリゲートが必要になる場合があります。

- サービスプロジェクト: レガシービジネスロジックのモダナイゼーション要素を含む

```
<business-app>-service.*.war
```

- ウェブプロジェクト: ユーザーインターフェイス関連の要素のモダナイゼーションが含まれます

```
<business-app>-web.*.war
```

ステップ 3: テスト環境を設定する

テスト環境を設定するには

1. モダナイズされたアプリケーションランタイムを設定します。詳細については、「AWS Mainframe Modernization ユーザーガイド」の「[ランタイムの設定](#)」セクションを参照してください。

Note

ランタイムコンポーネント固有の設定例については、AWS「自動リファクタリング開発環境のセットアップ」ガイドのセクション 5 を参照してください。

2. モダナイズされたアプリケーション用に入出力 (I/O) データセットを準備します。モダナイズされたアプリケーションは、シーケンシャル I/O データセット、VSAM データセット、その他を処理する場合があります。

Note

例については、「自動リファクタリング開発環境のセットアップ AWS」のセクション 6 を参照してください。

3. ランタイム環境 - 既存のランタイム環境を使用するか、新しいランタイム環境を作成できます。
 - 非マネージドランタイム環境を設定するには、[「非マネージドアプリケーションのセットアップ」](#)を参照してください。
 - マネージドランタイム環境を設定するには、[「マネージドアプリケーションのセットアップ」](#)を参照してください。

テスト環境を設定したら、モダナイズされたアプリケーションをデプロイする次のステップに進みます。

ステップ 4: モダナイズされたアプリケーションをデプロイする

[the section called “ステップ 2: モダナイズされたアプリケーションを構築する”](#) および [the section called “ステップ 3: テスト環境を設定する”](#) セクションで作成および/または設定したランタイムにアプリケーションアーティファクトをデプロイします。

モダナイズされたアプリケーションをデプロイするための追加のガイダンスは、次のリンクから確認できます。

- [Amazon EC2 にデプロイする](#)
- [Amazon ECS および Amazon EKS のコンテナにデプロイする](#)
- [AWS Mainframe Modernization アプリケーションを作成する](#)

ステップ 5: モダナイズされたアプリケーションをテストする

デプロイ後、

1. モダナイズされたアプリケーションを操作する方法については、利用可能な[ランタイム APIs](#)を確認してください。
2. アプリケーションをテストして、機能同等性をレガシーアプリケーションと一致させます。例えば、「AWS Mainframe Modernization ユーザーガイド」の[「サンプルアプリケーションをテストする」](#)を参照してください。

その他の例

AWS Transform を使用してメインフレームアプリケーションをモダナイズする具体的な例については、「[CardDemo メインフレームアプリケーションのモダナイズ](#)」を参照してください。

フルスタックの Windows モダナイゼーション

AWS Transform は、.NET モダナイゼーションや SQL Server モダナイゼーションなど、フルスタックの Windows モダナイゼーションを提供します。

Transform AWS agent for .NET は、クロスプラットフォーム .NET と互換性を持つように .NET アプリケーションをモダナイズするのに役立ちます。.NET [ウェブアプリまたは .NET IDE を使用して .NET](#) をモダナイズできます。

AWS Transform for [SQL Server モダナイゼーション](#)は、Microsoft SQL Server データベースおよび関連する .NET アプリケーションの Amazon Aurora PostgreSQL へのフルスタックモダナイゼーションを自動化する AI を活用したサービスです。このサービスは、スキーマ変換、データ移行、新しいターゲット PostgreSQL に合わせてアプリケーションコードを変更する移行ジャーニー全体を調整し、複雑で手間のかかるタスクを自動化することでチームの生産性を向上させます。

トピック

- [AWS 変換による .NET のモダナイズ](#)
- [SQL Server のモダナイゼーション](#)

AWS 変換による .NET のモダナイズ

.NET の AWS 変換エージェントは、クロスプラットフォーム .NET と互換性を持つように .NET アプリケーションをモダナイズするのに役立ちます。この機能は .NET モダナイゼーションと呼ばれます。AWS 変換[AWS 変換環境](#)の後、.NET モダナイゼーション変換ジョブを作成できます。

エクスペリエンス

AWS 変換は、いくつかのエクスペリエンスで利用できます。

- ウェブコンソール: 一度に最大 500 リポジトリの大規模な変換用。
- Visual Studio IDE: エージェントとインタラクティブかつ反復的に連携するソリューションまたはプロジェクトの開発者主導の変換。中規模から大規模のプロジェクトに推奨されます。

ウェブコンソールと IDE を自由に切り替えて、変換ジョブを操作できます。

他の IDEs や AI コードコンパニオンから AWS 変換を呼び出すこともできます。

- Kiro: Transform for Kiro パワーを使用して Kiro から AWS 変換します。

- その他の AI コードコンパニオン: Transform MCP エージェントを使用して、任意の AI コーディングアシスタントから AWS 変換します。

機能と主要機能

- ソース管理システムから .NET Framework コードベースを分析します。これには、プライベート NuGet サポート、クロスリポジトリの依存関係の特定、分析レポートの提供が含まれます。
- レガシー .NET Framework アプリケーションのクロスプラットフォーム .NET、E メール通知、変換概要レポートへの自動変換。
- ソースコントロールプラットフォーム (BitBucket、GitHub、GitLab) と簡単に統合して、既存のコードを取り込み、変換されたコードを新しいブランチにコミットします。
- ユニットテストによる変換されたコードの検証。

サポートされているバージョンとプロジェクトタイプ

AWS Transform は、以下のバージョンの .NET の変換をサポートしています。

- .NET Framework 3.5 以降
- .NET Core 3.1
- .NET 5.x+
- .NET 8

AWS 変換は、次のターゲット .NET バージョンに変換できます。

- .NET 8
- .NET 10

AWS Transform は、これらのタイプのプロジェクトの変換をサポートします (C# のみ)。

- クラスライブラリ
- コンソールアプリケーション
- ASP.NET:
 - フロントエンドのカミソリビューを含むモデルビューコントローラー (MVC)
 - 単一ページアプリケーション (SPA) バックエンド (ビジネスロジックレイヤー)

- Web API
- ウェブフォーム
- ユニットテストプロジェクト (NUnit、xUnit、MSTest)
- Windows Communication Foundation (WCF) サービス
- サードパーティーまたはプライベート NuGet パッケージ用にクロスプラットフォームバージョンが提供されているプロジェクト。クロスプラットフォームに相当するものがないか、使用できない場合、AWS Transform for .NET はベストエフォート変換を試みます。

AWS 変換では、次のプロジェクトタイプを最新の .NET に変換することもできます。これはプレビュー機能であり、サポートされているプロジェクトタイプほど完全に変換されない場合があります。

- WinForms デスクトッププロジェクト。
- WPF デスクトッププロジェクト。
- Xamarin モバイルプロジェクト。
- VB.NET で記述されたプロジェクト。

制限事項

AWS 変換のクォータと制限の詳細については、「」を参照してください[AWS 変換のクォータ](#)。

AWS 変換は以下を変換しません。

- Blazor UI コンポーネント
- コア互換ライブラリを持たない Win32 DLLs
- ソリューションを含まないリポジトリ。

AWS 変換は元のリポジトリブランチを変更せず、変換プランで指定された別のターゲットブランチにのみ書き込むことができます。

人間による介入

.NET Framework アプリケーションのクロスプラットフォーム .NET への移植中に、以下のシナリオで入力または承認の提供を求められる場合があります。

- ソースコードとアクセス許可へのコネクタの設定

- 提案されたモダナイゼーション計画を検証する
- 欠落しているパッケージの依存関係を NuGets としてアップロードする
- 変換されたコードを確認して受け入れる

詳細情報

AWS 変換ウェブアプリケーションまたは AWS Toolkit for を使用して、.NET コードをモダナイズできます Visual Studio。

- [AWS Transform ウェブアプリケーションを使用して .NET コードをモダナイズする](#)
- [IDE での .NET のモダナイズ](#)
- [.NET 変換のベストプラクティス](#)

AWS Transform ウェブアプリケーションを使用して .NET コードをモダナイズする

AWS Transform for .NET は、レガシー .NET アプリケーションをモダナイズするために設計された新しい生成 AI を活用したエージェントです。Transform ウェブアプリケーションまたは Visual Studio AWS Toolkit 拡張機能を使用して、レガシー .NET AWS コードをモダナイズできます。

Transform ウェブアプリケーションを使用して .NET AWS コードをモダナイズするには、以下の詳細なステップに従ってウェブアプリケーションに移動します。

1. [AWS Transform .NET ジョブプランの作成](#)
2. [ソースコードリポジトリコネクタの作成](#)
3. [変換の準備のためのリポジトリの確認](#)
4. [変換を準備するためのパッケージの依存関係の解決](#)
5. [トランスフォーメーションに備えるための計画の確認](#)
6. [.NET コードの変換](#)

AWS Transform .NET ジョブプランの作成

ワークスペースを作成したら、ジョブタブでジョブの作成を選択します。次に、自然言語を使用してチャットペインの AWS 変換からのプロンプトに従います。以下は、.NET モダナイゼーションジョブを作成するための一般的なステップです。

1. AWS Transform は、作成する変換ジョブのタイプを尋ねます。チャットで、.NET モダナイゼーションを入力します。
2. AWS 変換はジョブ名を提案し、ジョブ名を変更するかどうかを尋ねます。ジョブ名を変更する場合は、自然言語で AWS 変換を指示します。たとえば、ジョブ名を ExampleCorpDotNet1 に変更します。それ以外の場合は、チャットで提案されたジョブ名を受け入れることができます。ジョブ名を受け入れると、AWS Transform はジョブを作成していることをチャットウィンドウで通知します。
3. AWS Transform は変換ジョブを作成します。

AWS Transform .NET ジョブプランのコンポーネント

AWS Transform .NET ジョブディスプレイには、左端の垂直アイコンから選択できる 5 つのタブがあります。タスク、ダッシュボード、承認、アーティファクト、作業ログです。各タブには、左側のペインと中央のチャットペインがあります。追加の詳細を表示したり、人による入力をリクエストしたりするために、適切なコラボレーションペインが表示されることがあります。

タスク (ジョブプラン)

このタブには、左側のペインにジョブプランが表示されます。.NET ジョブには 5 つのステップがあります。

1. 変換するリソースを取得する: このフェーズでは、AWS CodeConnections を使用してコードリポジトリへのコネクタを作成します。リポジトリのアクセス許可によっては、コードリポジトリの管理者がコネクタを承認し、AWS Transform にリポジトリへのアクセスを許可する必要がある場合があります。
2. 変換用のリソースを検出する: このフェーズでは、AWS Transform はソースコントロール内のリポジトリを検出し、評価のために一部またはすべてを選択します。
3. 変換のコードを評価する: 選択したリポジトリが評価され、評価レポートを表示できます。
4. 変換の準備: このフェーズでは、リポジトリに依存関係がない場合は AWS、Transform から通知されます。欠落している依存関係をアップロードすることも、無視することもできます。リポジトリの管理者でない場合は、管理者が最終的な変換計画を承認する必要がある場合があります。
5. 変換: このフェーズでは、AWS Transform はリポジトリを変換し、変換が完了するまでの継続的なステータスを提供します。変換レポートを確認して、変更された内容と理由を理解できます。

各ステップのステータスを確認できます。

- 未開始

- ユーザー入力を待機する
- 進行中
- 完了

ダッシュボード

ダッシュボードタブには、変換の概要が表示されます。変換されたジョブの数、適用された変換、および変換の完了までの推定時間のメトリクスが表示されます。ダッシュボードの下には、リポジトリとそのステータス - 進行中、失敗、成功の表があります。

承認

ジョブの承認リクエストがこのタブに表示され、完了します。

アーティファクト

Jjob 関連のアーティファクトは、このタブからアップロードまたはダウンロードされます。

作業ログ

AWS Transform は、Worklog タブにそのアクションを記録します。Worklog は、AWS Transform が実行するアクションと、人間による入力リクエスト、およびそれらのリクエストに対するレスポンスの詳細なログを提供します。

ソースコードリポジトリコネクタの作成

の後[AWS Transform .NET ジョブプランの作成](#)、タスクタブで、AWS 変換ウィンドウの左ペインに変換ジョブのフェーズが一覧表示されます。最初のフェーズは、変換するリソースの取得です。このフェーズでは、共同作業者を招待し、ソースコードリポジトリに接続できます。AWS 変換ウィンドウの右ペインには、詳細を指定します。

各変換ジョブには、ソースコードリポジトリコネクタが 1 つだけ必要です。AWS Transform エージェントは、このコネクタを使用して、[AWS CodeConnections](#) を使用して GitHub、GitLab、または Bitbucket から .NET コードベースをダウンロードします。AWS 変換ジョブと同じリージョンに AWS CodeConnections を設定する必要があります。または、[ソースコードリポジトリの代わりに Amazon S3 のソースコードに接続](#)することもできます。

Note

既存のソースコードリポジトリコネクタがある場合、AWS Transform から通知されます。既存のコネクタを使用してこのコネクタを使用するを選択します。既存のコネクタを

使用しない場合は、[既存のコネクタの削除](#)の前に [新しいコネクタの追加](#)。ジョブごとに 1 つのソースリポジトリコネクタのみを持つことができます。

新しいコネクタの追加

新しいソースコードリポジトリコネクタを作成するには:

1. AWS CodeConnections コネクタに使用する AWS アカウント番号を入力します。
2. 同じ AWS アカウントに AWS CodeConnections を設定していない場合は、[Set up AWS CodeConnections](#) を使用します。
3. [AWS デベロッパーツールコンソール](#)を使用して、[Bitbucket](#)、[GitHub](#)、[GitHub Enterprise Server](#)、[GitLab](#)、または [AzureDevOps](#) リポジトリへの接続を作成します。
4. 作成した接続の Amazon リソースネーム (ARN) をコピーします。
5. AWS 変換に戻り、作成した接続の ARN を貼り付けます。
6. コネクタの名前を入力します。

Note

コネクタ名の一部として個人情報を入力しないでください。

7. コネクタの作成を開始するを選択します。
8. AWS Transform は、現在の変換ジョブと同じリージョンに、入力した AWS アカウントのコネクタを作成するリクエストを行い、承認リクエストが AWS マネジメントコンソールで AWS 管理者によって承認される準備ができていることを通知します。検証リンクをコピーを選択します。
9. AWS 管理者の場合は、AWS アカウントにサインインし、検証リンクに移動して接続リクエストを承認します。AWS 管理者でない場合は、検証リンクを AWS 管理者に提供します。
10. AWS 管理者がコネクタリクエストを承認したら、コネクタを確定を選択してコネクタ作成プロセスを完了します。別のコネクタを使用する場合は、再起動を選択してコネクタ作成プロセスを再起動します。

コラボレーションタブには、以下を含むコネクタの詳細も含まれています。

- ステータス -- 承認済みまたは保留中の承認
- AWS アカウント ID
- AWS リージョン

- 接続 ARN

既存のコネクタの削除

Note

既存のソースコードリポジトリコネクタがある場合、AWS Transform から通知されます。既存のコネクタを使用するか、削除を選択して新しいコネクタを作成します。

既存のソースコードリポジトリ接続を削除する場合、AWS Transform はコネクタを実際に削除する前に警告します。

1. 既存のコネクタは、いくつかの方法で削除できます。
 - a. AWS Transform からプロンプトが表示された場合は、削除を選択して新しいコネクタを作成できます。
 - b. プロンプトで再起動を選択することもできます。このコネクタを変更するには、再起動する必要があります。
2. AWS Transform は、再起動するとコネクタが削除されることを警告します。コネクタを削除するには、再起動を再度選択します。
3. AWS Transform は、コネクタを削除すると Bitbucket、GitHub、GitLab などのサードパーティリポジトリへの接続が削除されることを再度警告します。また、このコネクタを使用している AWS 変換ジョブは、コネクタが削除されると失敗します。削除を確認するには、「削除」と入力し、「削除」を選択します。
4. 新しいソースコードリポジトリコネクタを追加するには、「」を参照してください [新しいコネクタの追加](#)。

Amazon S3 のソースコードに接続する

ソースコードリポジトリを接続する代わりに、Amazon S3 で AWS Transform をソースコードに接続できます。 <https://docs.aws.amazon.com/transform/latest/userguide/dotnet-creating-repo-connector.html>

S3 バケットの整理

元のソースコードと変換されたコードは、次に示すように整理された共通の S3 バケットに保存されます。AWS 変換ジョブと同じリージョンに S3 バケットを設定する必要があります。ソースコード

をルートレベルで zip ファイルとしてバケットにアップロードします。zip ファイルには、各リポジトリの最上位フォルダが含まれている必要があります。

変換中、AWS Transform は変換出力フォルダを作成し、変換結果をそのフォルダに保存します。変換は、変換されたコードを含む transformed-code.zip という名前の zip ファイルを作成します。これには、プロジェクトレベルでファイルの変更を強調表示するコードの違いレポートファイル名 diff.txt が含まれます。

```
<customer-bucket>/
### source-code.zip
    ### repo 1
    ### repo 2
    ### .....
    ### repo n
### transform-output/
    ### transformed-code.zip
```

新しい S3 コネクタの追加

新しい S3 ソースコードリポジトリコネクタを作成するには:

1. AWS コンソールで、S3 バケットを作成します。
2. ソースコードを zip ファイルとして S3 バケットにアップロードします。
3. AWS 変換ウェブアプリケーションで、.NET 変換ジョブを開始します。ソースコードリポジトリの接続ステップで、S3 バケットでソースコードを接続する」を選択し、「保存」を選択します。
4. 次のページで、S3 バケットの詳細を入力し、送信を選択します。
 - a. コネクタ名
 - b. AWS アカウント ID
 - c. S3 バケット ARN
 - d. S3 バケット暗号化キー (オプション)
5. コネクタを承認します。
 - a. 次のページで、検証リンクをコピーします。
 - b. 承認者にリンクを参照させ、AWS 変換コネクタの作成リクエストページにアクセスします。
 - c. 新しいロールを作成するか、既存のロールを使用するかを選択します。
 - d. リクエストを確認したら、保存と承認を選択して承認します。

6. S3 zip ファイルの場所を指定します。
 - a. ウェブアプリの AWS 変換で、ステータスが「承認済み」と表示されるのを待ちます。
 - b. アセットの場所を指定ページで、コード zip ファイルの S3 URL を 形式で入力 `s3://bucket-name/zip-file-name` し、変換に送信を選択します AWS。
 - c. ジョブは Discovery ステップに進み、変換を続行します。

S3 コネクタの削除

既存の S3 コネクタを削除するには:

1. ウェブアプリの AWS 変換で、右上にあるコネクタの管理を選択します。
2. コネクタの管理ウィンドウで、コネクタを選択します。
3. [削除] を選択します。

変換の準備のためのリポジトリの確認

の後に [ソースコードリポジトリコネクタの作成](#)、変換の準備の最初のステップとしてリポジトリを確認します。以下の操作を実行できます。

- 計画の内容 [デフォルト設定](#) に影響するレビューと変更。
- 検出されたリポジトリを確認し、変換を評価するリポジトリを選択します。
- [リポジトリ評価レポート](#) を確認し、その内容について AWS Transform とチャットします。
- [Transform によって生成されたプランを使用する AWS](#)。
- [変換プランをカスタマイズする](#) ウェブアプリを使用するか、JSON ファイルをダウンロードして変更し、AWS 変換にアップロードします。

デフォルト設定

コラボレーションタブのジョブの詳細セクションで、以下を設定できます。

タスクタブの「変換の準備」 - 変換するリポジトリの確認ステップで、以下を設定できます。

- 変換されたコードがリポジトリに書き込まれるターゲットブランチ名を設定します。
- ターゲットの .NET バージョンを設定します。
 - .NET 8: すべてのプロジェクトを .NET 8 に変換する
 - .NET 10: すべてのプロジェクトを .NET 10 に変換する

- デフォルトのジョブ設定を更新するか、そのままにしておくことができます。具体的には次のとおりです。
 - 変換プランから .NET Standard プロジェクトを除外する この設定はデフォルトで選択されています。選択すると、AWS Transform は .NET Standard プロジェクトを変換プランから除外します。この設定を選択解除すると、.NET Standard プロジェクトが変換され、.NET Framework との互換性がなくなります。
 - NuGet ソースを確認し、.NET 互換パッケージバージョンを取得 AWS 変換が変換するコードの .NET 互換パッケージバージョンを検索して取得できるようにします。

リポジトリ評価レポートを確認する

リポジトリ評価レポートには、AWS 変換によって検出された各リポジトリに関する情報の表形式ビューが表示されます。これには、以下が含まれます。

- リポジトリ所有者
- 評価されたブランチ
- ソリューションの数
- プロジェクトの数
- コード行の合計
- 検出された .NET バージョン
- 検出されたプロジェクトタイプ
- プライベート Nuget 依存関係の数
- パブリック Nuget 依存関係の数
- 変換の複雑さ

ファイルをダウンロードするには:

1. ワークスペースで、コラボレーションタブを選択します。
2. リポジトリの選択ペインで、ダウンロードを選択し、HTML としてダウンロードまたは JSON としてダウンロードを選択します。

リポジトリの概要レポートに関するチャット

リポジトリの概要が非常に長い場合があります。チャットを使用して、その内容をよりよく理解します。Worklog タブにこのメッセージが表示されると、レポートがチャットで利用可能になります。評

価レポートは、クエリのチャットで利用可能になりました。チャットを開くには、ウェブコンソールの右下隅にある紫色の六角形アイコンをクリックします。プロンプトの例を次に示します。

- 検出または評価されたすべてのリポジトリを一覧表示します。
- 評価で検出された .NET 6 プロジェクトの数
- どのプロジェクトが最も複雑ですか？

Transform によって生成されたプランを使用する AWS

変換が成功する確率の高いリポジトリの検出から作成された AWS 変換プランを使用することを選択できます。これには、最後に変更されたリポジトリ、ルートリポジトリ、およびクロス依存リポジトリが含まれます。

AWS Transform には、変換プランに関する次のような概要が一覧表示されます。

- 検出されたリポジトリの合計数 Transform がソースコードリポジトリコネクタを使用して検出したレガシー .NET AWS リポジトリの数です。AWS Transform は、ジョブごとに最大 1,000 個のリポジトリをスキャンできます。スキャンまたは変換するリポジトリが他にもある場合は、複数の変換ジョブを作成できます。
- 選択したリポジトリ Transform が変換用に AWS 選択したレガシー .NET リポジトリの数。
- 選択した依存関係 選択したリポジトリに必要な依存関係の数。AWS Transform は自動検出され、変換プランに追加されます。

選択したリポジトリテーブルには、選択したリポジトリが一覧表示されます。リポジトリを名前で検索したり、リポジトリの追加ページに移動したり、選択したリポジトリとその依存関係を一覧表示する JSON 形式のファイルをダウンロードしたりできます。このビューでは、任意の変更を行うことができます。

選択済みリポジトリテーブルには、AWS 変換用に選択した変換リポジトリに関する次の詳細が一覧表示されます。

- リポジトリ名
- ソースブランチ
- 最終更新日時
- コード行 これにより、クォータ制限に近づいているかどうかを確認できます。詳細については、「[AWS 変換のクォータ](#)」を参照してください。

- 依存リポジトリ 依存リポジトリの数をクリックすると、これらの依存関係の詳細を表示できます。

AWS 変換生成変換プランに満足している場合は、リポジトリの確認を選択します。

AWS 変換生成変換プランを変更する場合は、 を選択します [変換プランをカスタマイズする](#)。

変換プランをカスタマイズする

を選択しない場合は [Transform によって生成されたプランを使用する AWS](#)、次のいずれかのオプションを使用して、変換するリポジトリのリストをカスタマイズできます。

1. 選択済みリポジトリテーブルでリポジトリを選択します。
 - a. リポジトリをプランに追加すると、AWS Transform はリポジトリの依存関係を選択し、自動的に変換プランに追加します。
 - b. テーブル内のリポジトリの選択を解除することもできます。
2. AWS 変換で生成されたリポジトリとブランチの JSON リストをダウンロードします。
 - a. ダウンロードを選択し、JSON としてダウンロードを選択します。
 - b. JSON を確認して変更します。変換しないリポジトリまたはリポジトリ内のソリューションを削除できます。
 - c. カスタマイズされたプランをアップロードを選択して、改訂された JSON をアップロードします。
 - d. JSON が正常にアップロードされると、選択したリポジトリテーブルに選択した内容が表示されます。
3. 依存リポジトリを表示を選択して、AWS Transform が検出したリポジトリのリストを、選択したリポジトリの依存関係として表示できます。
4. 必要に応じて、これらのステップを繰り返すことができます。カスタマイズされたプランに満足したら、リポジトリの確認を選択します。

変換を準備するためのパッケージの依存関係の解決

の後 [変換の準備のためのリポジトリの確認](#)、AWS Transform が欠落しているパッケージの依存関係を見つけた場合は、このステップを完了する必要があります。Windows PowerShell スクリプトを実行して、Visual Studio開発環境と同じデバイスから不足しているパッケージの依存関係を取得するか、不足しているパッケージを手動で取得できます。次に、不足しているパッケージをアップロードします。

欠落しているパッケージの依存関係は、次の 2 つの方法で更新できます。

- Artifact Connector を使用した解決: 専用の Connect Artifact Repository HITL ワークフローを使用して Artifact Repository (ADO NuGet Connector) を設定して解決します。これは、Artifact Connector が設定されておらず、パッケージがない場合に表示されます。
- 欠落しているパッケージの更新: AWS Transform は、欠落しているパッケージの依存関係テーブルに欠落しているパッケージを一覧表示します。検索ボックスで、見つからないパッケージを名前で検索できます。この表には、不足しているパッケージに関する以下の詳細が含まれています。
 - 名前
 - 関連付けられたリポジトリ
 - フレームワークバージョンのステータス
 - コアバージョンのステータス

欠落しているパッケージの依存関係を解決するには、[不足しているパッケージをアップロードする](#)。

不足しているパッケージをアップロードする

1. 選択すると、Windows PowerShell ヘルパースクリプトをダウンロードして、Visual Studio 開発環境内から不足しているパッケージの依存関係を取得できます。または、不足しているパッケージを手動で見つけることができます。

Windows PowerShell スクリプトを使用するには:

- a. Windows PowerShell スクリプトのダウンロードを選択します。
 - b. 欠落しているパッケージ依存関係ファイルを含むリポジトリへのアクティブな接続を使用して、スクリプトをローカルで実行します。
 - c. このスクリプトを使用すると、不足しているパッケージの依存関係をローカル環境にダウンロードできます。
2. このスクリプトは、1 つのアーカイブ内のすべての依存関係を含む、アップロードできる 1 つの zip ファイルを作成します。依存関係ごとに個別のファイル .nupkg または zip ファイルをアップロードすることもできます。
 3. パッケージファイルのアップロードを選択します。
 4. 依存関係ファイルのアップロードモーダルで、ファイルの選択を選択し、デバイス上の圧縮された欠落パッケージファイルの場所を参照します。
 5. アップロードを選択します。

6. AWS 変換は、アップロードしたファイルを検証します。検証中、更新を行うことはできません。AWS Transform は、欠落パッケージの依存関係テーブルの上に検証ステータスを報告します。
7. AWS Transform は、欠落パッケージの依存関係テーブルのステータス列も欠落から解決済みに更新します。パッケージが検証に失敗すると、そのステータスは Invalid になります。無効なファイルの場合は、次の操作を行います。
 - a. 欠落パッケージの依存関係テーブルで、チェックボックスを使用して無効なパッケージを選択します。
 - b. アップロードされたファイルの削除を選択します。
 - c. これにより、ステータスが欠落に戻ります。
8. 不足しているパッケージをアップロードし、パッケージの依存関係を解決したら、続行を選択して確認します。

欠落しているパッケージの依存関係を解決せずにレビューに進むを選択すると、AWS Transform は欠落しているパッケージなしで変換ジョブを開始するかどうかを尋ねます。欠落しているパッケージの依存関係を無視することを選択した場合、AWS Transform はアセンブリ参照を使用してコードを変換します。このアクションを続行すると、関連リソースに影響する可能性があります。

フォルダ構造の要件

NuGet アップロードの場合、AWS 変換には次のいずれかが必要です。

- 手動でアップロードされたパッケージのルートレベルに配置される .nuspec ファイル、または
- 提供された PowerShell スクリプトを使用して正しい構造を生成する

バージョンマッチングの要件

- フレームワークステータス: AWS 変換は、.csproj ファイル (ソースコード) で指定された依存関係のバージョンと、アップロードされた依存関係のバージョンとの一致を試みます。完全に一致すると、成功ステータスになります。
- コアステータス: AWS Transform は、アップロードされた .nuspec ファイルの依存関係セクションでコアバージョンの仕様を想定しています。見つかった場合、コアステータスは Success です。

形式例:

```
<dependencies>
  <group targetFramework="net8.0" />
</dependencies>
```

一般的なバージョンは次のとおりです。

- .NET 5.0 (net5.0)
- .NET 6.0 (net6.0)
- .NET 7.0 (net7.0)
- .NET 8.0 (net8.0)

トランスフォーメーションに備えるための計画の確認

[変換の準備のためのリポジトリの確認](#)、およびの[後変換を準備するためのパッケージの依存関係の解決](#)、AWS アカウント管理者は変換計画を確認し、AWS Transform で承認する必要があります。

AWS Transform は、変換用に選択されたジョブのリポジトリ、依存リポジトリ、依存パッケージのリストを表示します。

変換計画の確認

AWS Transform は、変換用に選択されたジョブのリポジトリ、依存リポジトリ、依存パッケージのリストを表示します。

Note

AWS 変換は、変換プランごとに最大 100 個の依存関係とリポジトリを変換できます。

1. AWS アカウント管理者でない場合は、ジョブプランを確認し、承諾する場合は、承認のために送信を選択します。
2. AWS アカウント管理者の場合は、プランを確認し、準備ができたら、変換を開始するプランを承認する必要があります。ジョブプランを確認したら、次のいずれかを選択します。
 - a. 拒否 ジョブが AWS アカウント管理者ではないユーザーによって作成された場合は、ジョブの再起動をジョブ作成者に通知することをお勧めします。
 - b. 変換を承認して開始します。

ジョブレビューには、次の詳細が含まれます。

1. ジョブの概要 これには以下が含まれます。

- a. Transform が AWS 変換されたコードを配置するターゲットブランチ。
- b. ターゲットの .NET バージョン、.NET 8.0 または .NET 10。
- c. ジョブ設定:
 - i. .NET 標準プロジェクトを除外する
- d. 変換用に選択されたりポジトリの数
- e. 依存リポジトリの数
- f. プライベート NuGet パッケージの数
- g. ジョブのコード行の合計

2. 選択したリポジトリ 変換用に選択されたりポジトリです。MVC、Web、Windows Communication Foundation (WCF)、コンソール、クラスライブラリ、UI フレームワーク - Razor ページ、またはユニットテストパッケージのいずれかである必要があります。この表には、次の情報が含まれます。

- a. 名前
- b. ソースブランチ
- c. サポートされているプロジェクト
- d. コード行
- e. 検出されたプロジェクト
- f. スキップされたプロジェクト
- g. 検出された依存関係

3. 追加された依存リポジトリ 変換用に追加された依存リポジトリです。MVC、Web、Windows Communication Foundation (WCF)、コンソール、クラスライブラリ、UI フレームワーク - Razor ページ、またはユニットテストパッケージのいずれかである必要があります。この表には、次の情報が含まれます。

- a. 名前
- b. で必要
- c. ソースブランチ
- d. サポートされているプロジェクト
- e. コード行

- g. スキップされたプロジェクト
- 4. 依存パッケージ 変換用に追加された依存パッケージです。MVC、Web、Windows Communication Foundation (WCF)、コンソール、クラスライブラリ、UI フレームワーク - Razor ページ、またはユニットテストパッケージのいずれかである必要があります。この表には、次の情報が含まれます。
 - a. 名前
 - b. 関連付けられたリポジトリ
 - c. フレームワークバージョンのステータス
 - d. コアバージョンのステータス

.NET コードの変換

AWS アカウント管理者が を完了すると [トランスフォーメーションに備えるための計画の確認](#)、変換ジョブの Dashboard タブで変換の進行状況を表示できます。

Note

リポジトリと依存関係の変換に加えて、AWS Transform は完全に変換された (ビルドエラーなし) ユニットテストプロジェクトを実行できます。AWS Transform は変換前にユニットテスト結果にアクセスできず、変換後の結果のみを共有できます。その後、変換前にベースラインデータと変換後の結果を比較して、潜在的なギャップを把握できます。

右上隅に、次のいずれかの値を持つジョブステータスが表示されます。

- ユーザー入力の待機中
- 経過時間
- 実行中

次のアイコンも表示されます。

- 変換停止アイコン
- 更新アイコン
- 設定アイコン

ダッシュボードには、変換の概要が表示されます。変換されたジョブの数と適用された変換のメトリクス、および変換の完了までの推定時間が表示されます。

ダッシュボードには以下が含まれます。

1. 変換ジョブの詳細セクションには、以下を含む変換ジョブのデフォルト設定と詳細が一覧表示されます。
 - a. ターゲットブランチの送信先 コードを変換するために、AWS Transform は変換されたコードの新しいブランチをコードリポジトリに作成します。
 - b. ターゲット .NET バージョン、.NET 8.0 または .NET 10
 - c. AWS 変換ジョブ ID。
 - d. ジョブ設定:
 - i. .NET 標準プロジェクトを除外する
2. 変換の概要セクションには、以下が含まれています。
 - a. 変換用に選択されたリポジトリの数
 - b. 変換するプロジェクトの数
 - c. これらのリポジトリとプロジェクトのコードの合計行数

変換が開始されると、円グラフがリポジトリステータス、パッケージステータス、ユニットテストステータスセクションに表示され、進行状況がリアルタイムで表示されます。ユニットテストのステータスは、変換後に AWS Transforms が実行するリポジトリ内のユニットテストのステータスを表示し、変換されたコードをテストします。AWS Transform は、実行されたテスト結果と、合格および不合格のユニットテストのリストを顧客が確認するための個々のテスト名を共有します。

Note

リポジトリセクションには、AWS 変換が推奨するリポジトリまたは変換用に選択したリポジトリが一覧表示されます。JSON のダウンロード を選択して、変換プランのリポジトリ、依存関係、パッケージのリストをダウンロードします。

変換レポートを確認する

変換ジョブが完了したら、[変換レポート](#)をダウンロードすることができます。

- [変換の概要レポート](#): このレポートには、HTML または JSON 形式の変換の概要が表示されます。ダッシュボードタブから変換の概要レポートをダウンロードします。Download を選択し、Download as HTML または Download as JSON を選択します。
- [変換詳細レポート](#): このレポートは、HTML 形式の変換に関する詳細情報を提供します。ダッシュボードタブから変換詳細レポートをダウンロードします。リポジトリリストで詳細レポートのダウンロードリンクを選択して、詳細レポートをダウンロードします。

変換レポートに関する AWS 変換とチャットする

リポジトリが完全に処理された後、または変換ジョブが完了した後に、変換レポート AWS について変換とチャットできます。Worklog タブには、チャットで利用可能な各リポジトリについてこのメッセージが表示されます。クエリのチャットでリポジトリ `repository_name` 変換の詳細を使用できるようになりました。変換ジョブが完了すると、このメッセージが表示されます。AWS 変換ジョブが完了しました。

チャットを開くには、ウェブコンソールの右下隅にある紫色の六角形アイコンをクリックします。

プロンプトの例を次に示します。

- どのプロジェクトが正常に変換されましたか？
- どのプロジェクトが部分的に移植されましたか？
- `repo_name` リポジトリにどのような変更が行われましたか？
- `project_name` プロジェクトではどのパッケージがアップグレードされましたか？

IDE での .NET のモダナイズ

AWS Transform には、AWS Toolkit for にインタラクティブなエージェント AI アシスタントが含まれています Visual Studio。これにより、IDE で会話型の step-by-step のガイド付きエクスペリエンスを通じてアプリケーションをモダナイズできます。.NET ソリューションまたはプロジェクトを変換するために、AWS 変換エージェントはコードベースを分析し、アプリケーションの移植に必要な更新を決定し、変換を開始する前に評価レポートと変換計画を生成します。

Visual Studio IDE エクスペリエンスは、IAM Identity Center による認証のみをサポートします。詳細については、「[IAM アイデンティティセンターへのユーザーの追加](#)」を参照してください。

IDE エクスペリエンスは、自律モードとインタラクティブモードを提供します。自動モードは、夜間または長時間実行されるジョブに適した無人変換に使用します。インタラクティブモードを使用して、チャットでエージェントと連携し、トランスフォーメーション、カスタマイズ可能なトランス

フォーメーションプラン、各プロジェクトのstep-by-stepレビューの可視性を高めます。繰り返して結果を絞り込んだり、満足するまで方向を変更したりできます。

変換ジョブがアクティブになると、次の 3 つのウィンドウを操作できます。

- チャットウィンドウは、変換エージェントとの主なインタラクションポイントです。チャットを使用して、質問、オプションの説明、変換のステアリング、反復を行います。
- AWS ジョブプランの変換ウィンドウには進行状況ステップが表示され、アーティファクトをダウンロードまたはアップロードしたり、コード変更の違いを表示したりできます。
- Worklog ウィンドウには、変換アクティビティのログが自然言語で表示されます。

AWS Transform は、.NET アプリケーションを Linux に移植するための 4 つの主要なタスクを実行します。

- .NET および言語バージョンのアップグレード – 古い .NET C# コードを現在の .NET および言語バージョンにアップグレードします。
- .NET Framework からクロスプラットフォーム .NET への変換 – Windows 依存の .NET Framework から Linux と互換性のあるクロスプラットフォーム .NET にプロジェクトとパッケージを変換します。
- Linux 互換のためのコードの書き換え – 非推奨および非効率的なコードコンポーネントをリファクタリングして書き換えます。
- Linux 互換性準備状況レポートの生成 – Linux でコードを構築して実行するためにユーザーの介入が必要なオープンエンドタスクの場合、AWS Transform は変換後にアプリケーションを設定するために必要なアクションの詳細なレポートを提供します。

AWS Transform が .NET 変換を実行する方法の詳細については、「」を参照してください[AWS Transform が .NET アプリケーションをモダナイズする方法](#)。

IDE からの AWS 変換を使用して .NET Visual Studio コードをモダナイズするには、以下を参照してください。

- [での AWS 変換を使用した .NET のモダナイズ Visual Studio](#)
- [AWS Transform が .NET アプリケーションをモダナイズする方法](#)
- [IDE での .NET 変換に関する問題のトラブルシューティング](#)

クォータ

IDE の AWS 変換 .NET 変換クォータについては、「」を参照してください[AWS 変換のクォータ](#)。

での AWS 変換を使用した .NET のモダナイズ Visual Studio

Windows ベースの .NET アプリケーションを Linux 互換のクロスプラットフォーム .NET アプリケーションに移植するには、2026 年または 2022 Visual Studio 年の AWS Transform を使用して以下の手順を実行します。

ステップ 1: 前提条件

続行する前に、Visual Studio Marketplace から [AWS Amazon Q で Toolkit 拡張機能](#) をダウンロードしてインストールしていることを確認してください。.NET モダナイゼーションの AWS 変換[機能と制限](#)を確認して、コードを変換できることを確認します。

ステップ 2: Visual Studio で認証する

Toolkit for Visual Studio から AWS アカウントに接続するには、AWS Toolkit ユーザーインターフェイスの開始方法 (接続 UI) を開きます。

1. Visual Studio のメインメニューから Extensions > AWS Toolkit > Getting Started に移動し、Getting Started: AWS Toolkit ページを開きます。
2. AWS 「変換認証」を選択します。
3. 既存のプロファイルを選択するか、新しいプロファイルを作成します。新しいプロファイルを作成するときは、プロファイルの名前と AWS 変換開始 URL を入力します。この URL は、管理者が IAM Identity Center に追加した後に E メールで送信されます。
4. 接続 を選択します。プロンプトに従ってサインインし、ウェブブラウザ経由でアクセス許可を付与します。Visual Studio では、「Connected with IAM Identity Center」と表示されます。
5. サインインすると、AWS 変換ダッシュボードに移動します。このランディングページから、以下を実行できます。
 - AWS 変換プロファイルの選択 (リージョン)
 - ワークスペースを選択または作成する
 - 変換ジョブのステータスを表示する

ステップ 3: アプリケーションを変換する

.NET ソリューションまたはプロジェクトを変換するには、次の手順を実行します。

変換を開始する

1. 変換Visual Studioする で C# または VB.NET ベースのソリューションまたはプロジェクトを開きます。
2. Solution Explorer から、変換するソリューションまたはプロジェクトを右クリックし、Port with AWS Transform を選択します。
3. Port with AWS Transform ダイアログが表示されます。
 - a. 「ワークスペースの選択」ドロップダウンで、ワークスペースを作成するか、既存のワークスペースを選択します。WorkSpaces は AWS 、変換ウェブアプリケーションにアクティビティを表示できるようにします。このアプリケーションでは、変換ジョブを表示およびチャットするように共同作業者を招待できます。
 - b. [.NET ターゲットの選択] ドロップダウンメニューで、アップグレードする .NET バージョンを選択します。
 - c. 変換モードでは、無人変換の場合は自律、インタラクティブ変換の場合はインタラクティブを選択します。
 - d. 開始 を選択して変換を開始します。
4. 変換ジョブが開始されると、AWS 変換ジョブプラン、チャット、および作業ログウィンドウが表示されます。

評価と変換計画

1. AWS Transform がコードを分析すると、評価レポートと変換計画がチャットに表示されます。インタラクティブモードでは、チャットは確認のために一時停止します。チャットで確認しない場合は、次の代替方法を使用できます。
 - コードエディタで評価または計画を表示するには、AWS Transform Job Plan ウィンドウで評価を表示または変換計画を表示を選択します。
 - 評価または計画をダウンロードするには、ジョブ計画ウィンドウのダウンロードアイコンを使用します。
2. 必要に応じて、変換プランを変更します。他のオプションを調べるなど、プランについてチャットで話し合うことができます。計画を変更するには、チャットで指示を与えるか、ジョブプランウィンドウのダウンロードアイコンを使用して改訂された変換計画をアップロードします。ステップは自由に追加、削除、または変更できます。チャットは、改訂された計画で指示を理解したことを確認します。

3. オプションでステアリングドキュメントをアップロードします。チャットウィンドウで、アップロードアイコンを使用して、アプリケーション仕様、組織の技術スタックの設定と規則、コーディング例などのステアリングドキュメントをアップロードします。ステアリングドキュメントをアップロードしたら、チャットウィンドウで説明して、エージェントにコンテキストを提供します。
4. プランに満足したら、チャットで承認すると、変換が開始されます。
5. インタラクティブモードでは、レビューのためにプロジェクトが変換された後にチェックポイントを停止するように設定できます。ジョブプランウィンドウのコード変換のステップには、ラウンドチェックポイントアイコンがあります。すべてのプロジェクトでは、チェックポイントがデフォルトで有効になっています。チェックボックスを使用してチェックポイントをクリアまたは設定します。

変換とチェックポイントのレビュー

1. AWS 変換はコードの変換を開始します。Transform AWS Job Plan ウィンドウで変換ステップの進行状況をモニタリングできます。ジョブプランウィンドウには、推定最大残り時間とジョブ ID が表示されます。
2. プロジェクトが変換されると、概要がチャットに表示され、コードの変更が適用されます。
3. インタラクティブモードでは、チェックポイントに到達すると、エージェントは一時停止して結果を確認できます。ジョブプランウィンドウでは、一時停止中のステップの結果を表示ボタンが表示され、コードの違いを確認できます。変更を確認したら、チャットでさらに変更をリクエストしたり、自分でコードを変更したりできます Visual Studio。
4. 必要に応じてチェックポイントで反復できます。満足したら、続行するようにチャットに指示すると、次のプロジェクトに進みます。

変換の終了

1. プロジェクトを変換すると、.NET エージェントはローカルビルドをトリガーし Visual Studio、コードがローカルマシンで正常にビルドされることを確認します。エラーが発生した場合は、エラーを解決するために機能します。
2. 変換が完了すると、変換タスク AWS Transform が実行できなかったり、Linux との互換性に必要な変更など、追加のステップが必要になる場合があります。次のステップのマークダウンファイル (NextSteps.md) をダウンロードして、AI コードコンパニオンへのプロンプトとして使用できます。ジョブプランウィンドウのダウンロードアイコンボタンを使用して、NextSteps.md をダウンロードします。

3. チャットで引き続きやり取りして、質問したり、追加の変更をリクエストしたりできます。
4. 変換ジョブを終了する準備ができたなら、チャットに伝えます。

AWS Transform が .NET アプリケーションをモダナイズする方法

Transform を使用した .NET AWS 変換の仕組みの詳細については、以下のセクションを参照してください。

アプリケーションの分析と変換計画の生成

変換を開始する前に、AWS Transform はコードをローカルに構築して、ビルド可能で変換用に正しく設定されていることを確認します。コードがビルドされない場合、AWS Transform は変換を続行するかどうかの入力を求めます。AWS Transform は次に、安全で暗号化されたビルド環境にコードをアップロードし AWS、コードベースを分析し、アプリケーションを移植するために必要な更新を決定します。

この分析中、AWS Transform は .NET ソリューションまたはプロジェクトをコードグループに分割します。コードグループは、ダイナミックリンクライブラリ (DLL) や実行可能ファイルなどのビルド可能なコード単位と一緒に生成するプロジェクトとそのすべての依存関係です。変換するすべてのプロジェクトの依存関係を選択していなくても、AWS Transform は選択したプロジェクトの構築に必要な依存関係を決定し、それらを変換して、変換されたアプリケーションが構築可能で使えるようにします。

コードを分析すると、AWS Transform は、変換されるコードグループとその依存関係のリストなど、提案された変更の概要を示す変換計画を生成します。

アプリケーションの変換

変換を開始するには、AWS Transform は安全なビルド環境でコードを再度ビルドし、リモートでビルド可能かどうかを確認します。AWS Transform は、アプリケーションの移植を開始します。最下位レベルの依存関係から順に処理します。AWS 変換が依存関係の移植で問題が発生した場合、変換を停止し、エラーの原因に関する情報を提供します。

変換には、アプリケーションに対する以下の変更が含まれます。

- 古い C# バージョンのコードを Linux 互換の C# バージョンに置き換える
- 以下を含む、.NET Framework のクロスプラットフォーム .NET へのアップグレード:
 - パッケージ、ライブラリ、API の特定と繰り返しの置き換え

- NuGet パッケージと API のアップグレードと置き換え
- クロスプラットフォームランタイムへの移行
- ミドルウェアのセットアップとランタイム設定の更新
- プライベートパッケージまたはサードパーティーパッケージの置き換え
- IIS および WCF コンポーネントの処理
- ビルドエラーのデバッグ
- 既存のコードを変換するための非推奨および非効率的なコードのリファクタリングや書き換えなど、Linux との互換性のためのコードの書き換え

変換レポートの確認と変更の受け入れ

変換が完了すると、AWS Transform は、変更されたファイル数、更新されたパッケージ数、変更された APIs 数など、アプリケーションに対して提案された更新に関する情報を含む変換レポートを提供します。影響を受けるファイルやファイルの一部、ビルドの試行中に発生したエラーなど、失敗した変換にフラグを付けます。また、ビルドログを使用してビルド概要を表示し、どのような変更が行われたかを確認することもできます。

変換レポートには Linux 移植ステータスも表示され、アプリケーションの Linux 互換性を確保するために追加のユーザー入力が必要かどうかを示します。コードグループ内のいずれかの項目で入力が必要な場合は、ビルド時に AWS Transform が対処できなかった Windows 固有の考慮事項を含む Linux 準備状況レポートをダウンロードします。コードグループまたはファイルに対して入力が必要な場合は、レポートを確認して、行う必要がある変更の種類の詳細を確認し、該当する場合は、コードの変更方法に関する推奨事項を確認します。これらの変更は、Linux でアプリケーションを実行する前に手動で行う必要があります。

ファイルへのインプレース更新として受け入れる前に、差分ビューで行われた AWS 変換の変更案を確認できます。ファイルを更新し、Linux 準備状況レポートの項目に対処すると、アプリケーションはクロスプラットフォーム .NET で実行できるようになります。

AWS 変換または AI コードコンパニオンを使用して、継続的な変換を求めるプロンプトを含む次のステップのマークダウンファイルをダウンロードできます。ファイルをダウンロードするには、レポートの右上にある次のステップのダウンロードボタンを選択します。

IDE での .NET 変換に関する問題のトラブルシューティング

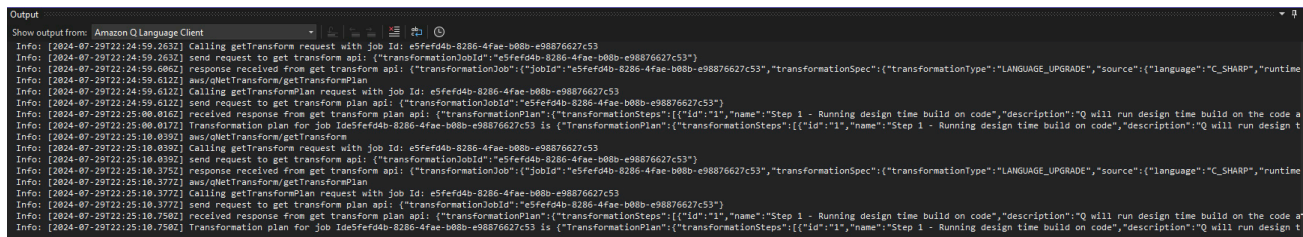
以下のセクションでは、AWS 変換を使用した IDE での .NET 変換に関する一般的な問題のトラブルシューティングを行います。

ジョブの進行状況を確認するにはどうすればよいですか？

Worklog ウィンドウに定期的なアクティビティが表示されます。Transform が AWS Transform AWS Job Plan ウィンドウのステップに長い時間を費やしているように見える場合は、出力ログでジョブがまだアクティブかどうかを確認できます。診断メッセージが生成されている場合、ジョブはまだアクティブです。

出力を確認するには、 の出力タブを選択しますVisual Studio。[次のソースの出力を表示:] メニューで、[Amazon Q 言語クライアント] を選択します。

次のスクリーンショットは、変換中に AWS Transform が生成する出力の例を示しています。



一部のプロジェクトが変換対象として選択されていないのはなぜですか？

AWS 変換は、サポートされている .NET プロジェクトタイプのみを変換できます。.NET プロジェクトを変換するためのサポートされているプロジェクトタイプとその他の前提条件の一覧については、「[ステップ 1: 前提条件](#)」を参照してください。

プロジェクトまたはソリューションが変換されていない場合、サポートを受けるにはどうすればよいですか？

自分で問題をトラブルシューティングできない場合は、または AWS アカウント チームに連絡してサポート サポートケースを送信できます。

サポートを受けるには、失敗したジョブを AWS が調査できるように、変換ジョブ ID を指定します。変換ジョブ ID を検索するには、 の出力タブを選択しますVisual Studio。[次のソースの出力を表示:] メニューで、[Amazon Q 言語クライアント] を選択します。

ファイアウォールが変換ジョブに干渉しないようにするにはどうすればよいですか？

組織がファイアウォールを使用している場合、 の変換を妨げる可能性がありますVisual Studio。Node.js のセキュリティチェックを一時的に無効にして、変換の実行を妨げる原因をトラブルシューティングまたはテストできます。

環境変数 `NODE_TLS_REJECT_UNAUTHORIZED` は、重要なセキュリティチェックを制御します。`NODE_TLS_REJECT_UNAUTHORIZED` を「0」に設定すると、Node.js による不正な TLS/SSL 証明書の拒否が無効になります。つまり、次のようになります。

- 自己署名証明書が受け入れられます。
- 期限切れの証明書が許可されます。
- ホスト名が一致しない証明書が許可されます。
- その他の証明書検証エラーが無視されます。

プロキシが自己証明書を使用している場合は、`NODE_TLS_REJECT_UNAUTHORIZED` を無効にする代わりに、次の環境変数を設定できます。

```
NODE_OPTIONS = -use-openssl-ca
NODE_EXTRA_CA_CERTS = Path/To/Corporate/Certs
```

それ以外の場合は、`NODE_TLS_REJECT_UNAUTHORIZED` を無効にするためにプロキシが使用する CA 証明書を指定する必要があります。

Windows で `NODE_TLS_REJECT_UNAUTHORIZED` を無効にするには：

1. スタートメニューを開き、環境変数を検索します。
2. [システム環境変数を編集する] を選択します。
3. [システムのプロパティ] ウィンドウで、[環境変数] をクリックします。
4. [システム変数] で、[新規] を選択します。
5. [変数名] を `NODE_TLS_REJECT_UNAUTHORIZED` に設定し、[変数値] を 0 に設定します。
6. [OK] を選択して変更を保存します。
7. Visual Studio を再起動します。

UI レイヤーコードの移植

AWS Transform は、ASP.NET ウェブサイトの UI レイヤーを、Linux で実行できる ASP.NET Core と互換性のある UI フレームワークに移植できます。次の変換を使用できます。

- MVC Razor UI 移植
- Blazor UI へのウェブフォームの移植

MVC Razor UI 移植

Razor Views UI を使用する ASP.NET MVC プロジェクトは、ASP.NET Core の MVC Razor Views に移植でき、同じ UI フレームワークに残ります。Razor ビューは ASP.NET と ASP.NET Core で似ていますが、移植コードの変更を必要とする違いがあります。

- ASP.NET MVC Razor Views は System.Web.Mvc 名前空間を使用し、.NET Framework と密接に関連付けられています。
- ASP.NET Core Razor Views は、Microsoft.AspNetCore.Mvc、依存関係インジェクションが改善されています。
- ASP.NET Core には、タグヘルパーやビューコンポーネントなどの新機能があります。

AWS 変換は、Razor Views で一般的な互換性のないコンポーネントとコンストラクトを検出してポートします。

Blazor UI へのウェブフォームの移植

ASP.NET Web Forms UI には、ASP.NET Core に対応するものではありません。AWS Transform は Web Forms UI レイヤーコードを ASP.NET Core の Blazor Server に移植できます。ターゲット UI フレームワークの UI レイヤーコードが書き換えられます。

Web Forms から Blazor UI への移植を有効または無効にするには、UI ASP.NET Web Forms から Blazor への変換ジョブ設定を確認またはクリアします。

変換の詳細

Web Forms プロジェクトは、サーバー側のホスティングを使用して Blazor プロジェクトに変換されます。変換中に、次のプロジェクトとファイルの変更が行われます。

ファイル変換マッピング

から	まで	説明
.aspx、.ascx	*.razor	.aspx ページと .ascx カスタムコントロールが .razor ファイルになる
Web.config	appsettings.json	Web.config 設定が appsettings.json 設定になる
Global.asax	Program.cs	Global.asax コードが Program.cs コードになる

から	まで	説明
*.master	*layout.razor	マスターファイルが layout.razor ファイルになる

変換後のプロジェクト構造

変換後、変換されたファイルは次のプロジェクトの場所にあります。

- コンポーネントフォルダ: すべてのカミソリコンポーネント (*.razor) は、プロジェクトの .csproj ファイルと同じレベルで作成されたコンポーネントフォルダに配置されます。
- ページマッピング: Web Forms ページ (*.aspx) は Components/Pages フォルダ構造に変換されます。
- コントロールマッピング: Web Forms コントロール (*.ascx) は、Components フォルダに直接配置されます。

制限とサポートされていない機能

現在、以下の機能はサポートされていません。ワークロードがこれらの機能を使用する場合、そのコードは手動で移植するか、AI コードコンパニオンで変換後に移植する必要があります。

- .svc ファイル (サービスファイル)
- .ashx ファイル (汎用ハンドラーファイル)
- サービスリファレンスとウェブリファレンス (reference.map、reference.cs、.wsdl ファイル)
- 同じプロジェクト内の .aspx (Web Forms) ファイルと .cshtml (Razor) ファイルの混合
- AWS Transform for .NET でまだサポートされていないプロジェクトタイプの依存関係
- サードパーティーベンダーの UI ライブラリ/コントロール
- ウェブサイトプロジェクト

ウェブアプリケーションプロジェクト (.csproj ファイルを含む) のみがサポートされています。ウェブサイトプロジェクトをウェブアプリケーションプロジェクトに変換するには、「Microsoft guidance [Converting a Web Site Project to a Web Application Project](#)」を参照してください。

変換レポート

.NET 変換ジョブが完了すると、次の変換レポートを使用できます。

- [変換の概要レポート](#)
- [変換詳細レポート](#)

変換の概要レポート

変換ジョブが完了すると、ウェブコンソールで変換の概要レポートを使用できます。ダッシュボードタブから変換の概要レポートをダウンロードします。Download Transformation Summary Report を選択し、Download as HTML または Download as JSON を選択します。

変換概要レポートには、変換の概要と、変換された各リポジトリに固有の情報が表示されます。

- ジョブの詳細
 - 変換ジョブの概要
 - 変換結果の概要
 - ファイルの概要
- リポジトリリスト、および各リポジトリの:
 - リポジトリ名の概要
 - ソリューション変換の概要
 - ステータス、タイプ、.NET バージョン、コード行の合計、変換されたコード行、変更されたファイルを含むプロジェクトの詳細

変換詳細レポート

変換ジョブが完了すると、ウェブコンソールと IDE ユーザーの両方で変換の詳細レポートを使用できます。

- ウェブコンソールユーザー: 変換されたリポジトリごとに詳細レポートを使用できます。ダッシュボードタブで、リポジトリリストの詳細レポートのダウンロードリンクを使用して詳細レポートをダウンロードします。
- IDE ユーザー: ダウンロードアイコンを使用して、変換レポートをAWS 変換ジョブプランウィンドウからダウンロードします。

変換詳細レポートは、インタラクティブな HTML レポートです。ファイル変更の理由や実用的なエラーの詳細など、変換に関する詳細情報が含まれています。情報は、次の階層に配置されます。

- ジョブ情報

- 各ソリューションについて:
 - ソリューション変換の概要
 - 変換の概要
 - プロジェクト
- 各プロジェクトについて:
 - プロジェクトの概要
 - ビルドエラーの概要
 - NuGet パッケージの変更
 - ファイルの変更
 - ビルドエラー
- ユニットテスト結果
- Linux 準備状況レポート

例を含む変換詳細レポートセクションのチュートリアルについては、ブログ記事「[Announcing AWS Transform for .NET detailed transformation reports](#)」を参照してください。

.NET 変換のベストプラクティス

AWS Transform を使用して .NET モダナイゼーションの結果を向上させるには、このガイダンスに従ってください。これらのベストプラクティスは、現在 IDE で利用可能な .NET 会話 AI Visual Studio アシスタントに適用されます。

トピック

- [変換する前に](#)
- [評価と計画](#)
- [インタラクティブまたは自律的に変換する](#)
- [検証と確定](#)

変換する前に

.NET 変換ジョブを開始する前に、次の推奨事項を確認してください。

.NET 会話 AI アシスタントを使用する

最適な変換結果を得るには、新しい .NET エージェントを使用します。会話型 AI アシスタントは Toolkit [AWS for Visual Studio](#) Visual Studioを通じて で利用できます。また、AWS 変換および AWS 変換 MCP エージェントKiroの機能を通じて、[Kiro](#)または他の IDEs でも使用できます。新しい .NET エージェントは、ウェブコンソールではまだ利用できません。

目的が .NET から .NET への変換でない場合は、他のツールを使用します。

AWS .NET の変換は、.NET Framework から .NET コードのモダナイゼーション、および .NET からそれ以降のバージョンへのアップグレードに適しています。非 を含む変換には適していません。Java などの NET 言語、または 以外の言語。React などの NET フレームワーク。また、アプリケーションアーキテクチャのリファクタリングも提供しません。このようなモダナイゼーションKiro には、[AWS 変換カスタム](#)または を使用します。

安定版の Visual Studioを使用する、できれば VS2026

プレビューソフトウェアによる複雑さを避けるために、プレビューまたはインサイダーエディション Visual Studioではなく の GA エディションを使用します。Toolkit AWS は 2026 Visual Studio または 2022 で使用できます。VS2026 はメモリ消費量を減らすことでパフォーマンスが向上するため、推奨されます。

最新の AWS Toolkit for を使用する Visual Studio

AWS Toolkit for のバグ修正と機能更新Visual Studioの最新バージョンにとどまります。ツールキットのバージョンを確認し、拡張機能 > 拡張機能の管理 > インストール済みでアップグレードできます。

中程度以上の複雑さのプロジェクトで Visual Studio IDE を操作する

IDE エクスペリエンスは、.NET エージェントと連携してプロジェクトを変換できる、変換の最も深いビューを提供します。中程度、高度、または重大な複雑さのプロジェクトの場合は、ウェブコンソールではなく IDE を使用します。

検証計画を立てる

変換後にアプリケーションの正確性を検証する方法を決定します。変換後、実行時に動作して正しく見えること、およびセキュリティコントロール、ビジネスロジック、データストレージがそのままであることを確認する必要があります。検証は手動で、またはテストオートメーションを使用して実施されますか？ソリューションにユニットテストが含まれている場合、AWS Transform はそれらを移植して実行し、結果が変換レポートに表示されます。

NuGet プライベートパッケージの依存関係を提供する

ソリューションがプライベートパッケージに依存している場合は、IDE と AWS Transform が見つかるように、ローカルマシンで利用できることを確認してください。

パッケージの置き換えを指定するか、エージェントのレコメンデーションをリクエストする

最新の .NET バージョンがないパッケージの場合、変換プランをカスタマイズして目的のパッケージ置換を指定するか、エージェントにレコメンデーションを要求できます。

これ以上の変更が必要でないことが確実になるまで、ジョブを終了しないでください。

変換が完了したら、追加の変更が必要な場合に備えて、ジョブをすぐに終了しないでください。変換は、「」と言うまで完了しません。ジョブの完了をマークするまで、変更を要求できます。

変換後、AI コードコンパニオンに引き渡し、アプリを確定します。

AWS Transform を などの AI コードコンパニオンと組み合わせることを期待しますKiro。一般的なパターンは、最初に AWS Transform for .NET で変換してから AI コードコンパニオンに引き渡し、アプリケーションを確定することです。この確定作業には、ランタイムエラー、動作の問題、UX スタイルの特定と修正が含まれる場合があります。

モダナイゼーションジャーニーでは、AWS トランスフォームを目標に向かって長い道のりを進むエクスペストラベルとして考え、AI コードコンパニオンを最終目的地に向かうローカルトラベルとして考えてみましょう。

AWS Transform から AI コードコンパニオンへの引き渡しを支援するために、AWS Transform は変換の最後に HTML 変換レポートと Next Steps マークダウンドキュメントを提供します。

評価と計画

変換ジョブを開始すると、AWS Transform はコードを分析し、評価レポートと変換計画を作成します。このフェーズを最大限に活用するには、次のベストプラクティスを使用します。

評価を確認し、質問をして期待値を設定する

評価の結果を注意深く確認し、どのプロジェクトを変革するのが難しいのか、またその理由を理解します。エージェントは、複雑なレベル (低、中、高、重大) のプロジェクトを提示し、その評価を受ける理由を説明します。

難しい分野について理解を深めるために質問します。例えば、次のようになります。

- プロジェクト変換またはユースケースにおける信頼度をエージェントに尋ねます。
- AI コードコンパニオンで必要となる可能性のある変換後のタスクをエージェントに尋ねます。

プランを確認し、エージェントとオプションについて話し合う

エージェントは、レビュー用の変換プランを提供します。このプランについて話し合い、変更することができます。まず、オプションを理解します。プランの任意の部分に対してどのような代替案があるかエージェントに尋ねます。

たとえば、変換プランでは、WCF サービスを ASP.NET Web APIs。ただし、組織は WCF の使用を希望しています。エージェントにオプションを尋ね、代替手段としてに変換することを学びます CoreWCF。

ステアリングドキュメントをアップロードして組織の計画を調整する

元の変換計画は、組織の要件と好みを理解せずに作成されます。組織の規則、アプリケーションの仕様、推奨されるテクノロジースタック、推奨されるコーディングパターンの例をエージェントに通知できます。

ステアリングドキュメントをアップロードしたら、それを説明してエージェントコンテキストを提供します。ドキュメントのステアリングからの追加情報により、エージェントは組織の働き方に合わせた計画を準備できます。

プランを自由にカスタマイズする

.NET エージェントは柔軟性があり、.NET から .NET への変換のドメイン内で変換プランを自由にカスタマイズできます。チャットでプランの変更をリクエストすることも、必要に応じてマークダウンファイルを使用することもできます。プランを確認して、満たされるまで絞り込みます。

プランを複数の利害関係者がレビューする必要がある場合は、プランをマークダウンファイルとしてダウンロードして共有し、後でカスタマイズされたプランをアップロードできます。

たとえば、変換プランを確認し、クラスライブラリプロジェクトを .NET 10 ではなく .NET Standard に変換することを決定します。チャットで、「すべてのクラスライブラリを .NET Standard 2.0 に変換する計画を改訂する」と言うと、エージェントは改訂された計画を提示します。

インタラクティブまたは自律的に変換する

Visual Studio IDE エクスペリエンスは、変換中のインタラクティブ度を制御するための自律モードとインタラクティブモードを提供します。

インタラクティブモードと自律モードを使用してインタラクティブ度を制御する

変換を開始するときにモードを選択し、チャットウィンドウの上部にあるドロップダウンからいつでも変更できます。

- 自動モードは、夜間または長時間実行されるジョブに適した無人変換に使用します。引き続き結果を確認し、ジョブの最後に変更を求めることができます。
- インタラクティブモードを使用して、カスタマイズ可能な変換計画、変換の詳細の高度な可視性、各プロジェクトのstep-by-stepレビューにより、変換プロセス全体でチャットでエージェントと連携します。繰り返して結果を絞り込んだり、満足するまで方向を変更したりできます。

モードを組み合わせることができます。たとえば、インタラクティブモードで開始して変換計画を確認して調整し、自律モードに切り替えてプロジェクトを自動的に変換します。

エージェントにコード変更の説明を求める

コードの変更がわからない場合は、エージェントに説明を求めてください。

チェックポイントを使用して、レビューまたは変更のためにプロジェクトを停止する

インタラクティブモードでは、計画を承認した後、プロジェクトごとにチェックポイントを設定またはクリアできます。チェックポイントを持つプロジェクトが変換されると、変換されたコードが IDE のプロジェクトに適用され、エージェントはレビューのために一時停止します。結果を確認でき、気に入らない場合は、さらに変更を求めることができます。

チェックポイントで停止している間は、次のことができます。

- チャットで表示された結果を確認し、チャットで質問をします。例：「ファイル X で暗号化 API を変更したのはなぜですか？」
- AWS 変換ジョブプランウィンドウでプロジェクトの結果を表示ボタンを選択して、コードの変更を表示します。
- 追加の変更をリクエストして、変換を絞り込みます。例：「ロガー X を使用するように変更する」または「パッケージ Y の代わりにパッケージ X を使用するように変更する」。
- 変換を再試行し、別の手順でやり直します。例えば、「このクラスライブラリを .NET 10 ではなく .NET Standard をターゲットに変更する」または「この変換を再試行し、ターゲットを から MVC Razor Pages Blazor に変更する」です。
- IDE で独自のコード変更を行います。

- 次のプロジェクトチェックポイントに進むようにエージェントに指示します。

満足するまで反復する

モダナイゼーションは本質的に反復的であり、エージェントで反復することが期待できます。プロジェクトチェックポイント、および変換の終了時に変更を加えることができます。必要に応じて、満足するまで繰り返します。変換は、「」と言うまで完了しません。

チェックポイントで停止したときにエージェントと一緒にコーディングする

エージェントを操作するときに、独自のコード変更を行うことができます。チェックポイントで停止すると、でコードファイルを編集できます Visual Studio。続行するようにエージェントに指示すると、変更は同期されます。

検証と確定

変換が完了したら、変換されたアプリケーションを検証し、本番稼働用に確定します。

変換レポートと次のステップを確認して、何が実行され、何が残っているかを理解します。

HTML 変換レポートを確認して、何が実行されたかを理解します。次のステップのマークダウンファイルを確認して、残っているタスクを理解します。明確でない点を明確にするようエージェントに依頼します。

AI コードコンパニオンを使用してアプリケーションを検証、デバッグ、確定する

変換されたアプリケーションが正しいことを確認します。などの AI コードコンパニオン [Kiro](#) を使用して問題を解決します。

ユニットテストを検証する

ソリューションにユニットテストプロジェクト (NUnit、または MSTest) が含まれている場合 xUnit、AWS Transform はそれらを最新の .NET に移植し、テストを実行します。HTML 変換レポートでユニットテスト結果を確認します。AI コードコンパニオンを使用して、失敗したテストを修正します。

アプリケーションの機能を検証する

アプリケーションを実行し、その機能が元のものと一致することを確認します。機能の問題とランタイムエラーを書き留めます。AI コードコンパニオンで修正します。

アプリケーション UI を検証する

アプリケーションを実行し、ナビゲーションやルーティングなど、元のアプリケーションの外観と動作を確認します。アプリケーションにスタイルがない場合、.css ファイルが適切な場所がない可能性があります。AI コードコンパニオンで修正します。

SQL Server のモダナイゼーション

AWS Transform for SQL Server Modernization は、Microsoft SQL Server データベースおよび関連する .NET アプリケーションの Amazon Aurora PostgreSQL へのフルスタックのモダナイゼーションを自動化する AI を活用したサービスです。このサービスは、スキーマ変換、データ移行、新しいターゲット PostgreSQL に合わせてアプリケーションコードを変更する移行ジャーニー全体を調整し、複雑で手間のかかるタスクを自動化することでチームの生産性を向上させます。

サポート対象のリージョン

AWS Transform for SQL Server が米国東部 (バージニア北部) - us-east-1 で利用可能に

クロスリージョン使用状況: サポートされていないリージョンのデータベースの場合、データベースをサポートされているリージョンにクローンして変換し、結果をターゲットリージョンにデプロイできます。

機能と主要機能

データベース変換

- スキーマ変換: テーブル、ビュー、インデックス、制約、関係など、SQL Server スキーマを Aurora PostgreSQL に自動的に変換します
- ストアドプロシージャ変換: T-SQL ストアドプロシージャを AI で強化された精度で PL/pgSQL に変換します
- データ移行: AWS Database Migration Service (DMS) を使用して整合性検証でデータを移行する
- データベースオブジェクト: トリガー、関数、ビュー、計算列、および ID 列をサポート
- 検証: 自動データ整合性検証と参照整合性チェック

アプリケーション変換

- Entity Framework 変換: PostgreSQL の Entity Framework 6.3-6.5 および EF Core 1.0-8.0 設定を更新します

- ADO.NET 変換: SQL Server から PostgreSQL プロバイダーに ADO.NET データアクセスコードを変換します。
- 接続文字列の更新: すべてのデータベース接続文字列を新しいターゲット PostgreSQL データベースに自動的に更新します。
- データベースプロバイダーの変更: SQL Server プロバイダーを Npgsql (PostgreSQL プロバイダー) に置き換えます
- ORM 設定の更新: データ型マッピング、ID 列、およびデータベース固有の設定を変更します。

オーケストレーションと検証

- ウェーブベースのモダナイゼーション: 大規模な資産を論理的な移行フェーズに整理する
- 依存関係マッピング: アプリケーションとデータベース間の関係を識別します
- ヒューHuman-in-the-loop (HITL) チェックポイント: 重要な段階でレビューと承認のゲートを提供します
- 自動検証: スキーマの互換性、データ整合性、アプリケーション機能をテストします
- CI/CD 統合: 既存の開発パイプラインと統合

デプロイメント

- Amazon ECS および Amazon EC2 デプロイ: 自動スケーリングサポートによるコンテナ化された自動デプロイ
- Infrastructure-as-code 生成: CloudFormation テンプレートまたは AWS CDK テンプレートを作成します。
- 自動デプロイ検証: ヘルスチェックで正常なデプロイを検証します
- ロールバック機能: 問題が発生した場合のロールバック手順をサポートします

サポートされているバージョンとプロジェクトタイプ

SQL Server バージョン

AWS Transform は、次の SQL Server バージョンをサポートしています。

SQL Server バージョン	サポートステータス
SQL Server 2022	サポート
SQL Server 2019	サポート
SQL Server 2017	サポート
SQL Server 2016	サポート
SQL Server 2014	サポート
SQL Server 2012	サポート
SQL Server 2008 R2	サポート

 Note

SQL Server のすべてのエディションがサポートされています (Express、Standard、Enterprise)。SQL Server は、AWS (Amazon RDS for SQL Server または Amazon EC2 の SQL Server) でホストすることも、の外部でホストすることもできます AWS。

.NET バージョン

.NET バージョン	サポートステータス
.NET 10	サポート
.NET 8	サポート
.NET 7	サポート
.NET 6 (コア)	サポート
.NET Framework 4.x 以前	サポート外

⚠ Important

レガシー .NET Framework 4.x 以前のバージョンはサポートされていません。アプリケーションで .NET Framework を使用している場合は、SQL Server 変換機能を使用する前に、まず AWS Transform for .NET モダナイゼーションを使用して .NET Core 6+ にアップグレードする必要があります。

Entity Framework のバージョン

フレームワーク	サポートされるバージョン
エンティティフレームワーク 6	6.3、6.4、6.5
エンティティフレームワークコア	1.0 から 8.0
ADO.NET	すべてのバージョン (GA)

ソースコードリポジトリ

AWS Transform は、次のソースコードプラットフォームをサポートしています。

- GitHub および GitHub Enterprise Server
- GitLab.com と GitLab セルフマネージド
- Bitbucket クラウドと Bitbucket データセンター
- Azure DevOps と Azure DevOps Server
- Amazon S3

[Target database] (ターゲットデータベース)

AWS 変換は、最新の Aurora 機能と最適化をサポートする Amazon Aurora PostgreSQL (PostgreSQL 15 以降互換) をターゲットとしています。

技術要件

データベースの要件

- Microsoft SQL Server バージョン 2008 R2 ~ 2022
- (AWS Amazon RDS for SQL Server または Amazon EC2 上の SQL Server) でホストされている SQL Server、または の外部でホストされている SQL Server AWS
- AWSホスト型データベースの場合、データベースと AWS 変換は同じ AWS リージョンにある必要があります。
- の外部でホストされているデータベースの場合 AWS、AWS 変換サービスへのネットワーク接続が必要です。
- VIEW DEFINITION および VIEW DATABASE STATE アクセス許可を持つデータベースユーザー
- 印刷可能な ASCII 文字のみを使用するデータベースパスワード (「/」、「@」、「」、スペースを除く)
- ソース SQL Server を含む VPC には、少なくとも 2 つの異なるアベイラビリティーゾーンにサブネットが必要です (DMS レプリケーションサブネットグループで必要)

アプリケーション要件

- .NET Core 6、7、または 8 アプリケーション
- Entity Framework 6.3-6.5 または Entity Framework Core 1.0-8.0、または ADO.NET
- ソースコードで検出可能なデータベース接続
- アプリケーションが正常にビルドおよび実行される
- サポートされているリポジトリプラットフォームのソースコード

AWS アカウントの要件

- AWS 管理者アクセス権を持つ アカウント
- IAM アイデンティティセンターが有効
- 必要なサービスロールが作成されました (以下のセットアップ手順を参照)
- 適切なネットワーク設定を持つ VPC

データ処理とストレージ

処理場所

- スキーマ処理は VPC 内の DMS インスタンスで行われます
- データ移行はオプションであり、必要に応じて除外できます。
- 変換アーティファクトは AWS 変換サービスリージョンに保存されます

保存済みアーティファクト

以下の項目はサービスリージョンに保存されます。

- エージェントログ
- 評価結果
- SQL スキーマファイル
- DMS 出力アーティファクト

Important

データレジデンシーにとって重要: データ移行がオプトアウトされた場合でも、メタデータと処理アーティファクトはサービスリージョンに保存されます。これは、厳格なデータレジデンシー要件を持つ組織にとって重要です。

アーティファクト管理

- 独自の KMS キーを使用した暗号化のカスタマーオプション
- すべてのアーティファクトに定義された TTL (time-to-live) 期間
- アーティファクトはオフラインストレージ用にダウンロード可能

SQL Server の要件

データベースの要件

外部でホストされるデータベース

の外部でホストされているデータベース AWS がサポートされています。外部でホストされたデータベースを使用するには、データベースと AWS Transform サービス間のネットワーク接続を確認する必要があります。

SSIS ワークロード

- 制限: SQL Server Integration Services (SSIS) 変換はサポートされていません。
- 回避策: SSIS 移行には AWS Glue または他の ETL サービスを使用します。

リンクサーバー

- 制限: リンクサーバーを介した外部データベース接続には、人間による設定が必要です。
- 回避策: 移行後にリンクサーバーを再設定するか、データベースリンクやアプリケーションレベルの統合などの代替アプローチを使用します。

複雑な T-SQL パターン

- 制限: 一部の高度な T-SQL パターンでは、人間による介入が必要になる場合があります。
- 回避策: 複雑な SQL 変換に Amazon Kiro CLI を使用します。AWS 変換は、変換プロセス中にレビューするためにこれらにフラグを付けます。

アプリケーション要件

レガシー .NET Framework

- 制限: .NET Framework 4.x 以前のバージョンはサポートされていません。
- 回避策: AWS Transform for .NET を使用して最初に .NET Core 6+ にアップグレードしてから、SQL Server 変換を使用します。

Entity Framework のバージョン

- 制限: Entity Framework 6.3-6.5 と EF Core 1.0-8.0 のみがサポートされています。
- 回避策: 変換前にサポートされている Entity Framework バージョンにアップグレードします。

VB.NET アプリケーション

- 制限: VB.NET はサポートされていません。
- 回避策: C# に変換するか、AWS 変換カスタムを使用して VB.NET から C# に変換します。

クロスデータベースの依存関係

- 制限: データベーススキーマが複数のデータベース間でやり取りする際の課題。
- 回避策: 移行前にクロスデータベースクエリを確認してリファクタリングします。データベースの統合または PostgreSQL スキーマの使用を検討してください。
- 影響: 複雑なデータベース間のシナリオでは、人間による介入が必要になる場合があります。

リポジトリとデータベースの結合

- 制限: 1 つのリポジトリが複数のデータベースを提供する場合の課題。
- 回避策: リポジトリの再構築または段階的な移行アプローチを検討してください。
- 影響: ウェーブベースの移行には追加の計画が必要になる場合があります。

インフラストラクチャ要件

ジョブあたりの単一アカウント/リージョン

- 制限: 各変換ジョブは 1 つの AWS アカウントとリージョンを対象としています。
- 回避策: マルチアカウントまたはマルチリージョンデプロイ用に複数の変換ジョブを作成します。

デプロイターゲット

- 制限: Amazon ECS と Amazon EC2 のデプロイがサポートされています。

リポジトリの要件

プライベート NuGet パッケージ

- 制限: プライベート NuGet パッケージには追加の設定が必要です。
- 回避策: ジョブを開始する前に、変換設定でプライベート NuGet フィードを設定します。

SQL Server のセットアップ

SQL Server 環境でこれらのステップを実行して、AWS 変換モダナイゼーションを有効にします。

データベースのセットアップと設定

ステップ 1: 必要なアクセス許可を持つデータベースユーザーを作成する

必要なアクセス許可を持つ AWS 変換専用のデータベースユーザーを作成します。DMS Schema Conversion ユーザーがすでにある場合は、再利用できます。

SQL Server インスタンスに接続し、次のコマンドを実行します。

```
-- 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

モダナイズするデータベースごとに、データベース固有のコマンド (USE、CREATE USER、GRANT) を繰り返します。

db_datareader ロールは、スキーマ変換のみではなく、データ移行にのみ必要です。

- db_datareader ロールは、データベース内のすべてのテーブルへの読み取りアクセスを許可します。
- このロールは、データ移行を実行する場合にのみ必要です。
- スキーマ変換のみ (データ移行なし) の場合、db_datareader ロールは必要ありません
- スキーマ変換には、他のアクセス許可 (定義の表示、データベースの状態の表示など) で十分です。

ステップ 2: AWS Secrets Manager に認証情報を保存する

データベース認証情報を AWS Secrets Manager に安全に保存します。DMS 用に作成されたシークレットがすでにある場合は、このステップをスキップします。

1. コンソールで AWS Secrets Manager に移動する
2. 新しいシークレットの保存を選択する
3. シークレットを設定します。
 - シークレットタイプ: 他のデータベースの認証情報
 - データベース: Microsoft SQL Server
 - ユーザー名: atx_user (または選択したユーザー名)
 - パスワード: 作成したパスワード
 - サーバー名: SQL Server エンドポイント
 - データベース名: データベース名
 - ポート: 1433 (またはカスタムポート)
4. [次へ] を選択します。
5. シークレット名を入力: atx-db-modernization-sqlserver
6. 必要なタグを追加します (これらのタグは必須です) 。
 - キー: プロジェクト、値: atx-db-modernization

- キー: 所有者、値: データベースコネクタ
7. 残りの画面で次へを選択する
 8. ストアを選択する
 9. 次のステップで使用するシークレット ARN を書き留めます。

Important

データベースパスワードは、「/」、「@」、「」、スペースを除く、印刷可能な ASCII 文字のみを使用する必要があります。削除が予定されているシークレットは、変換の失敗を引き起こす可能性があります。

ステップ 3: 必要な DMS ロールを作成する

AWS 変換には、DMS オペレーションに特定の IAM ロールが必要です。以下の CloudFormation テンプレートを使用して、これらのロールをデプロイします。

Note

AWS アカウントに既存の DMS 関連のロールがある場合は、重複を作成するのではなく、このテンプレートを変更してそれらのリソースを再利用します。

次の内容で `dms-roles.yaml` という名前のファイルを作成します。

```
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

DMSecretsManagerRole:

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
```

CLI を使用して CloudFormation AWS スタックをデプロイします。

```
aws cloudformation create-stack \
  --stack-name dms-roles \
  --template-body file://dms-roles.yaml \
  --capabilities CAPABILITY_NAMED_IAM \
  --region us-east-1
```

または、AWS コンソールを使用してデプロイします。

1. AWS コンソールで CloudFormation に移動する
2. スタックの作成を選択する
3. テンプレートファイルのアップロードを選択する
4. dms-roles.yaml ファイルをアップロードする
5. スタック名の入力: dms-roles
6. IAM 機能の確認

7. スタックの作成を選択する

ステップ 4: ネットワークセキュリティを設定する

AWS Transform、SQL Server データベース、およびその他の AWS サービス間のネットワーク接続が適切であることを確認します。

セキュリティグループ設定 (推奨アプローチ)

推奨されるアプローチ: IP ベースのルールではなく、セキュリティグループベースのアクセスコントロールを使用します。これにより、セキュリティが向上し、管理が容易になり、AWS Transform のアーキテクチャとシームレスに連携します。

セキュリティグループベースのアクセスコントロールを使用する理由

- DMS Schema Conversion が VPC 内に Elastic Network Interface (ENIs)を作成する
- データベースはパブリックにアクセス可能である必要はありません。
- AWS 変換ではプライベート IP アドレスが公開されないため、IP ベースのルールが複雑になる
- セキュリティグループリファレンスは、リソースのスケールに応じて動的で自動的に更新を提供します。

SQL Server セキュリティグループを設定する

AWS 変換で DMS スキーマ変換インスタンスプロファイルを設定するときは、DMS SC インスタンスのセキュリティグループを指定します。データベースセキュリティグループは、この DMS SC セキュリティグループからのインバウンドトラフィックを許可する必要があります。

Step-by-step設定:

1. DMS Schema Conversion セキュリティグループを特定します。
 - これは、AWS 変換でインスタンスプロファイルを作成するときに指定されます
 - セキュリティグループ ID を書き留めます (例: sg-0123456789abcdef0)
2. SQL Server セキュリティグループのインバウンドルールを更新します。
 - タイプ: カスタム TCP
 - ポート: 1433 (またはカスタム SQL Server ポート)
 - ソース: DMS Schema Conversion セキュリティグループ ID
 - 説明: 「DMS Schema Conversion アクセスを許可する」

3. Aurora PostgreSQL ターゲットの場合 (作成後):

- タイプ: PostgreSQL
- ポート: 5432 (またはカスタム PostgreSQL ポート)
- ソース: DMS Schema Conversion セキュリティグループ ID
- 説明: 「DMS Schema Conversion アクセスを許可する」

Important

最小特権セキュリティモデルにとって重要: 組織がデフォルトですべてのトラフィックをブロックする「最小特権」セキュリティモデルを使用している場合は、DMS Schema Conversion Security Group からデータベースポートへのインバウンドトラフィックを明示的に許可する必要があります。すべてのソースまたは IP 範囲に対してポート 1433 を開かないでください。

必要な AWS サービス接続

VPC が以下と通信できることを確認します。

- AWS サービスエンドポイントの変換
- AWS DMS エンドポイント
- Aurora PostgreSQL エンドポイント
- アーティファクトストレージ用の S3 エンドポイント
- AWS Secrets Manager エンドポイント
- AWS CodeConnections エンドポイント

VPC エンドポイント: プライベートネットワークの場合は、インターネットゲートウェイの依存関係を避けるために、必要な AWS サービスの VPC エンドポイントを設定します。

外部でホストされるデータベースの要件

SQL Server データベースが の外部でホストされている場合は AWS、モダナイゼーションを開始する前に、次の前提条件が満たされていることを確認し、セットアップステップを完了してください。

前提条件

- VPC を持つ AWS アカウント

- VPC と外部データベース間のネットワーク接続。ネットワーク接続の設定については、AWS DMS 「ユーザーガイド」の「[ネットワーク接続の設定](#)」を参照してください。

セットアップ手順

1. 外部データベースの接続の詳細 AWS Secrets Manager を使用して、にシークレットを作成します。詳細については、「[the section called “ステップ 2: AWS Secrets Manager に認証情報を保存する”](#)」を参照してください。
2. プロンプトが表示されたら、外部データベースに接続するための VPC ID とセキュリティグループ ID を指定します。シークレットのデータベースホスト名は AWS アカウント内で解決できないため、AWS はこの情報の入力を求めます。

AWS 変換設定

Transform AWS コンソールにアクセスし、新しいモダナイゼーションプロジェクトを開始して、モダナイゼーションジャーニーを開始します。このステップでは、ワークスペースを確立し、フルスタック変換の基本的な設定をセットアップします。

完了するアクション:

- AWS マネジメントコンソールの AWS 変換サービスに移動する
- サービスオプションから Microsoft Modernization を選択する
- 新しいモダナイゼーションプロジェクトの作成を選択します。
- プロジェクト名と説明を入力します。
- ターゲット AWS リージョンを選択 (SQL Server の場所と一致する必要があります)
- プロジェクト設定の確認と確認

Note

ベストプラクティス: アプリケーション名とターゲット環境 (CustomerPortal-Production-Modernization など) を含むわかりやすいプロジェクト名を使用して、マルチウェーブシナリオでプロジェクトを簡単に識別します。

SQL Server モダナイゼーションジョブを作成する

データベースとソースコードリポジトリを接続する

データベースコネクタを設定する

データベースコネクタを設定して、SQL Server データベースへの安全な接続を確立します。コネクタは環境分析、依存関係検出を実行し、AWS Transform が評価と変換のためにデータベーススキーマとメタデータにアクセスできるようにします。

完了するアクション:

- セットアップウィザードからデータベースコネクタの設定を選択します。
- 接続方法の選択: 新しい接続または既存の接続
- SQL Server 接続の詳細 (エンドポイント、ポート、データベース名) を指定する
- 認証とテスト接続を設定する
- 検出されたデータベースを確認し、選択を確認する

Note

ネットワーク接続: SQL Server セキュリティグループと NACLs、AWS Transform サービスエンドポイントからのインバウンド接続が許可されていることを確認します。

ソースコードリポジトリを接続する

AWS Transform を有効にして .NET アプリケーションのソースコードにアクセスします。AWS Transform は、ソースコードを提供する 3 つの方法をサポートしています。Personal Access Token (PAT) コネクタ (推奨)、AWS CodeConnections、または Amazon S3。この統合により、サービスはアプリケーションコードの分析、データベースの依存関係の特定、PostgreSQL との互換性のための自動コード変換を実行できます。

完了するアクション:

- Connect Source Code Repository セクションに移動します。
- 認証方法を選択する: PAT Connector (推奨)、AWS CodeConnections、または Amazon S3
- .NET アプリケーションを含むリポジトリを選択する
- 分析用のブランチを指定する (通常は main/master または development)
- リポジトリのアクセスとコード構造を検証する

Note

リポジトリ検出: AWS Transform はリポジトリを自動的にスキャンして、.NET プロジェクト、エンティティフレームワーク設定、データベース接続文字列、SQL 依存関係を識別します。

Note

セキュリティ上の注意: AWS Transform は、リポジトリへの読み取りアクセスのみを必要とし、変換されたコード用の新しい機能ブランチを作成します。メインブランチは変更されません。

ループ内の人間

AWS トランスフォームは、ヒューman-in-the-loop (HITL) メカニズムを使用して品質を確保し、重要なトランスフォーメーションの決定を確認および承認できるようにします。次のチェックポイントには注意が必要です。

ウェーブプランのレビューと承認

レビュー内容: グループ化されているデータベースとアプリケーション、ウェーブのシーケンスなど、提案された移行ウェーブ。

以下の操作を実行できます。

- ウェーブプランを承認する
- ウェーブ間でデータベースを移動してウェーブをカスタマイズする
- ウェーブシーケンスの変更
- 波を分割またはマージする

承認後、AWS Transform はスキーマ変換に進みます。

スキーマ変換のレビュー

確認内容: テーブル、ストアドプロシージャ、関数、トリガーを含む変換されたデータベースオブジェクト。アクション項目では、注意が必要なオブジェクトが強調表示されます。

以下の操作を実行できます。

- 変換されたコードを受け入れる
- 変換されたコードを変更する
- 変換後の人間によるレビューのフラグ
- 元のコードと変換されたコードのside-by-side比較を表示する

承認後: AWS Transform はスキーマを Aurora PostgreSQL に適用し、データ移行を続行します (設定されている場合)。

アプリケーションコードのレビュー

確認内容: Entity Framework の設定、接続文字列、データアクセスコード、ストアドプロシージャ呼び出しなど、すべてのアプリケーションコードの変更。

以下の操作を実行できます。

- 各ファイルの変更を受け入れる
- 変換されたコードを変更する
- 変更を拒否する (非推奨)
- チームにコメントを追加する
- ローカルレビュー用に変換されたコードをダウンロードする

承認後: AWS Transform はリポジトリ内の新しいブランチに変更をコミットし、検証に進みます。

検証結果のレビュー

確認内容: スキーマの互換性、データ整合性チェック、アプリケーションビルドステータスなどの自動検証結果。

以下の操作を実行できます。

- 合格したテストを確認する
- 失敗したテストを調査する
- 警告に対処する
- デプロイに進むか、問題の修正に戻る

承認後: AWS Transform はターゲット環境にデプロイする準備をします。

デプロイの承認

確認内容: infrastructure-as-code テンプレート、ECS サービス設定、デプロイ設定などのデプロイ設定。

以下の操作を実行できます。

- デプロイ設定の確認とカスタマイズ
- デプロイを承認して続行する
- 追加テストのデプロイを遅らせる
- レビュー用のインフラストラクチャコードをダウンロードする

承認後、AWS Transform はモダナイズされたアプリケーションとデータベースをターゲット環境にデプロイします。

評価とウェーブプランニング

ランディングゾーンを設定する

モダナイズされたアプリケーションをデプロイするターゲットインフラストラクチャ環境を設定します。ランディングゾーンの設定には、Aurora PostgreSQL のプロビジョニング、ネットワーク設定、リファレンスデプロイ設定が含まれます。

完了するアクション:

- 設定メニューからランディングゾーンの設定を選択します。
- Aurora PostgreSQL ターゲットを設定する (バージョン、インスタンスクラス、マルチ AZ)
- ネットワークおよびセキュリティ設定 (VPC、サブネット、セキュリティグループ) を設定する

Note

コストに関する考慮事項: Aurora PostgreSQL インスタンスには継続的なコストが発生します。開発/テスト環境で Aurora Serverless v2 を使用することを検討してください。

評価

データベーススキーマ、アプリケーションコード、依存関係を包括的に分析します。評価フェーズでは、変換の複雑さの評価、潜在的な課題の特定、詳細なレポートの生成を行います。

完了するアクション:

- コード分析用のリポジトリブランチを選択する
- 変換するリポジトリとデータベースを選択する
- 評価結果と複雑さの評価を確認する
- エフォート射影を使用して評価レポートを生成する
- ステークホルダーのレビューと承認のためにレポートをエクスポートする

Note

評価インサイト: 評価は、スキーマオブジェクト、ストアドプロシージャ、アプリケーションの依存関係の詳細な分析を提供し、必要な人間の介入のレベルを推定します。

ウェーブプランニング

変換の複雑さ、ビジネスの優先順位、アプリケーションの依存関係に基づいて、モダナイゼーションを管理可能なウェーブに整理します。AWS Transform は、カスタマイズ可能なインテリジェントなウェーブレコメンデーションを生成します。

完了するアクション:

- AI 生成の Wave Plan とレコメンデーションを確認する
- ビジネスの優先順位に基づいて Wave 設定をカスタマイズする
- アプリケーションとデータベース間の依存関係を検証する
- 会話型インターフェイスを使用して Wave Plan を確定する
- ウェーブ実行の優先順位を設定し、責任を割り当てる

Note

AI を活用したレコメンデーション: 自然言語を使用して AI を操作し、CustomerDB をウェブ 1 に移動する」または「これら 2 つのアプリケーションを同じウェブに結合する」というレコメンデーションを調整できます。

Note

ベストプラクティス: 1~2 個の低複雑さデータベースを含むパイロットウェブから始めて、プロセスを確立し、チームの信頼を構築します。

各ウェブのデータ移行

各ウェブのスキーマ変換

AI で強化された変換機能を使用して、SQL Server データベーススキーマを PostgreSQL 互換の同等のスキーマに変換します。このプロセスは、テーブル、ストアドプロシージャ、関数、トリガー、およびその他のデータベースオブジェクトを変換します。

完了するアクション:

- スキーマ変換ターゲットを指定して設定する
- DMS Schema Conversion サービスを使用して自動変換を実行する
- 変換結果を確認し、人間の介入項目を特定する
- HITL プロンプトによる手動介入の処理
- 必要に応じて、人間による修正を検証し、変換を再実行します。

Note

一般的な手動介入: リンクサーバー、ユーザー定義タイプ、高度な T-SQL パターン、ベンダー固有の SQL 機能には、通常、人間によるレビューと修正が必要です。

各ウェブのデータ移行

AWS Database Migration Service (AWS DMS) 統合を使用して SQL Server から Aurora PostgreSQL にデータを転送します。テスト目的で、本番データ移行または合成データ生成を選択します。

完了するアクション:

- 本番データ移行または合成データ生成を選択する
- データ転送用に DMS レプリケーションインスタンスを設定する
- 移行の進行状況をモニタリングし、データの不整合を処理する
- データ整合性と参照制約を検証する
- 続行する前に正常に完了したことを確認する

Note

自動 DMS 統合: AWS 変換は、DMS 設定の複雑さを抽象化し、エンドポイントの作成、タスク設定、データ型マッピングを自動的に処理します。

Note

データ検証: サービスは、ソースデータベースとターゲットデータベースの両方に対して同一のクエリを実行して、データの整合性を確保することで、経験的テストを実行します。

各ウェブのコード移行

Entity Framework 設定の更新、接続文字列の変更、SQL クエリの適応、フレームワーク固有の調整など、.NET アプリケーションコードを Aurora PostgreSQL と連携するように変換します。

完了するアクション:

- SQL Server の依存関係の自動コード分析を実行する
- フレームワーク変換の実行 (エンティティフレームワークプロバイダーの更新)
- SQL クエリ構文を PostgreSQL 互換 SQL に適応させる
- 変換されたコードのターゲットブランチ管理を設定する
- 複雑なパターンの手動介入を処理する

SQL Server モダナイゼーションワークフロー

このセクションでは、AWS Transform を使用した SQL Server のモダナイゼーションプロセス全体をstep-by-stepで説明します。

ステップ 1: SQL Server モダナイゼーションジョブを作成する

Transform AWS コンソールで新しい変換ジョブを作成して、モダナイゼーションジャーニーを開始します。

1. AWS 変換コンソールにサインインする
2. モダナイゼーションジョブの作成を選択する
3. Windows モダナイゼーションジョブを選択し、SQL Server モダナイゼーションを選択します。
4. ジョブの詳細を入力します。
 - ジョブ名: プロジェクトの記述名
 - 説明: オプションの説明
 - デプロイのターゲット region: AWS region
5. [ジョブの作成] を選択します。

Important

ジョブ名に個人を特定できる情報 (PII) を含めないでください。

ステップ 2: SQL Server データベースに接続する

AWS 変換を SQL Server データベースに接続して、スキーマ分析と変換を有効にします。

データベースコネクタを作成する

1. SQL Server モダナイゼーションジョブで、リソースへの接続に移動します。
2. SQL Server データベースへの接続を選択する
3. 新しいコネクタの作成 を選択します。
4. コネクタ情報を入力します。
 - コネクター名: 記述名
 - AWS アカウント ID: SQL Server がホストされているアカウント

5. 確認されると、承認用のリンクが届きます。承認リンクをコピーして、アカウントの AWS 管理者から承認を取得します。承認したら、次のステップに進むことができます。
6. 管理者がコネクタリクエストを承認したら、送信をクリックしてソースコード接続のセットアップに進みます。

ステップ 3: ソースコードリポジトリを接続する

AWS Transform は、SQL Server データベースとやり取りするコードを分析および変換するために、.NET アプリケーションのソースコードにアクセスする必要があります。AWS Transform は、ソースコードを提供する 3 つの方法をサポートしています。

認証方法を選択する

Personal Access Token (PAT) コネクタ (推奨)

カスタムアクセス許可スコープ、セルフホストプロバイダーのサポート、または GitHub Secrets などのプロバイダー固有の APIs へのアクセスを必要とするチームに最適です。カスタムアクセス許可を持つソースコードプロバイダーに PAT を作成し、それを に保存し AWS Secrets Manager、必要に応じて AWS Transform が取得します。トークンのローテーションと有効期限を管理するのはお客様の責任です。

AWS CodeConnections

認証情報の自動管理を必要とするチームに最適です。AWS CodeConnections は、OAuth 2.0 認可フローを通じて認証を処理するマネージドプロバイダー統合を使用します。は、トークンの自動更新やローテーションなど、認証情報のライフサイクル全体 AWS を管理します。認証情報を手動で管理する必要はありません。

Amazon S3

ソースコードを Amazon S3 バケットに直接アップロードします。AWS Transform は、変換ジョブ中にバケットからコードにアクセスします。

機能	PAT コネクタ (推奨)	AWS CodeConnections
認証情報管理	手動 (カスタマー管理)	自動 (AWSマネージド)
トークンのライフサイクル	手動ローテーションが必要	自動更新

機能	PAT コネクタ (推奨)	AWS CodeConnections
アクセス許可の柔軟性	完全にカスタマイズ可能なスコープ	修正されたアクセス許可
セルフホスト型プロバイダーのサポート	サポート	利用不可
セットアップの複雑さ	中程度 (手動トークンの作成と保存)	低 (1 回限りの認可)
トークンストレージ	お客様の AWS Secrets Manager	AWS マネージド

PAT コネクタを設定する (推奨)

PAT コネクタを使用して、カスタムアクセス許可を持つ個人用アクセストークンをソースコードプロバイダーに作成し、安全に保存し AWS Secrets Manager、必要に応じて AWS Transform が取得します。ローテーションや有効期限などのトークンライフサイクルを管理する責任があります。AWS Transform は、シークレットにアクセスするためのアクセス許可を持つ必要な IAM ロールを自動的に作成します。

PAT コネクタは、セルフホスト型およびカスタム DNS/URL バージョンを含む次のプロバイダーをサポートしています。

- GitHub および GitHub Enterprise Server
- GitLab.com と GitLab セルフマネージド
- Bitbucket クラウドと Bitbucket データセンター
- Azure DevOps と Azure DevOps Server

個人用アクセストークンを作成する

ソースコードプロバイダーに PAT を作成します。必要なアクセス許可はプロバイダーによって異なります。プロバイダーのタブを選択します。

⚠ Important

作成直後にトークンをコピーします。再度表示することはできません。変換ジョブの有効期間を設定します。有効期限が期限切れにならないように設定しないでください。

⚠ Warning

PAT トークンをコードリポジトリにコミットしたり、安全でないチャネルで共有したりしないでください。常に に保存します AWS Secrets Manager。

GitHub

設定、デベロッパー設定、個人用アクセストークン、きめ細かなトークンに移動します。変換するリポジトリを選択し、次のアクセス許可を付与します。

リポジトリのアクセス許可

アクセス許可	アクセス	目的
内容	読み取りと書き込み	ソースコードを読み取り、変換されたコードをリポジトリに書き込みます
メタデータ	[Read-only]	基本的なりポジトリ情報にアクセスします

組織のアクセス許可 (組織リポジトリに必要)

アクセス許可	アクセス	目的
メンバー	[Read-only]	リポジトリ検出のためにトークンにアクセスできる組織を一覧表示します

GitLab

プロファイルの編集、トークンへのアクセスに移動します。次のスコープを選択します。

スコープ	目的
read_api	リポジトリメタデータ、プロジェクト情報、ユーザーの詳細を読み取り、グループとブランチを一覧表示します
read_repository	分析用のソースコードファイルとリポジトリ構造を読み取ります
write_repository	変換されたコードをリポジトリに書き込みます

Bitbucket

アカウント設定、セキュリティ、API トークンの作成と管理に移動します。必要なスコープはトークンタイプによって異なります。

Workspace/リポジトリトークン (ATCT — ベアラー認証、ユーザー名不要)

アクセス許可	アクセス	目的
リポジトリ	読み取りと書き込み	リポジトリを一覧表示し、ブランチを読み取り、git push を介して変換されたコードを書き込みます

アカウント API トークン (ATAT — E メールによる基本的な認証) またはアプリパスワード (ATBB — ユーザー名による基本的な認証)

スコープ	目的
read:account	認証されたユーザーを識別してリポジトリメンバーシップを解決します
read:workspace:bitbucket	Transform がリポジトリを AWS 列挙できるように、トークンがアクセスできるワークスペースを一覧表示します。シークレットでワークスペースリストを指定する場合は必須ではありません。

スコープ	目的
read:repository:bitbucket	リポジトリを一覧表示し、メタデータとブランチ情報を読み取ります
write:repository:bitbucket	変換されたコードを git push 経由でリポジトリに書き込みます

Azure DevOps

ユーザー設定、個人用アクセストークンに移動します。カスタム定義スコープを選択します。組織のスコープでは、アクセス可能なすべての組織 (推奨) を選択するか、単一の組織を指定します。

スコープ	アクセス	目的
コード	読み取りと書き込み	ソースコードを読み取り、リポジトリとブランチを一覧表示し、変換されたコードを書き戻します
ユーザープロフィール	読み取り	トークンアクセスを検証し、組織検索用のユーザー ID を検出します
メンバー権限管理	読み取り	リポジトリ検出のためにトークンにアクセスできる組織を一覧表示します

PAT を に保存する AWS Secrets Manager

1. AWS Secrets Manager コンソールを開きます。
2. 新しいシークレットを保存 を選択します。
3. [Secret type] (シークレットタイプ) で、[Other type of secret] (他の種類のシークレット) を選択します。
4. プロバイダーとホスティングタイプに基づいてキーと値のペアを追加します。
 - クラウドホストプロバイダー — PAT を値として token という名前のキーを追加します。
 - 特定の組織を持つ Azure DevOps の場合は、組織名organizationで という名前のキーも追加します。

- Bitbucket アプリパスワード (ATBB) の場合は、Bitbucket ユーザー名usernameで という名前のキーも追加します。Bitbucket アカウント API トークン (ATAT) の場合は、Bitbucket メールアドレスemailで という名前のキーを追加します。
- セルフホスト型およびカスタム DNS/URL プロバイダー — (hostサーバー URL、例: `https://github.mycompany.com`)、`provider_type` (github、gitlab、)bitbucket、`token` (PATado) の各キーを追加します。
- 特定の組織を持つ Azure DevOps の場合は、組織名organizationで という名前のキーも追加します。
- Bitbucket アプリパスワード (ATBB) の場合は、Bitbucket ユーザー名usernameで という名前のキーも追加します。Bitbucket アカウント API トークン (ATAT) の場合は、Bitbucket メールアドレスemailで という名前のキーを追加します。

次の例は、シークレットが GitHub クラウドホストプロバイダー AWS Secrets Manager を検索する方法を示しています。

```
{
  "token": "your-github-personal-access-token"
}
```

次の例は、Bitbucket アプリパスワード (ATBB) を示しています。

```
{
  "token": "your-bitbucket-app-password",
  "username": "my-bitbucket-username"
}
```

次の例は、セルフホスト型 GitLab インスタンスを示しています。

```
{
  "host": "https://gitlab.mycompany.com",
  "provider_type": "gitlab",
  "token": "your-gitlab-personal-access-token"
}
```

5. [次へ] を選択します。
6. シークレット名を入力します。例: `github-pat-myproject`。
7. (オプション) 暗号化するカスタマーマネージド KMS キーを選択します。

8. ウィザードを完了し、保存を選択します。
9. シークレット ARN をコピーします。この値は、AWS 変換ジョブを設定するときに必要になります。

カスタマー管理の KMS キーを使用してシークレットを暗号化する場合は (デフォルト AWS 管理のキーではなく)、AWS トランスフォームがシークレットを復号できるように KMS キーポリシーを更新する必要があります。カスタマー管理の KMS キーポリシーに次のステートメントを追加します。

```
{
  "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"
    }
  }
}
```

REGION を自分の AWS リージョン (など us-east-1) に置き換え、*YOUR-SECRET-ARN* をシークレットの ARN に置き換えます。この kms:ViaService 条件により、KMS キーは AWS Secrets Manager サービスを介してのみ使用できます。この kms:EncryptionContext:SecretARN 条件は、復号を特定のシークレットに制限します。

KMS キーポリシーを更新するには:

1. で AWS KMS コンソールを開きます <https://console.aws.amazon.com/kms>。
2. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
3. KMS キーを選択します。
4. アクセスポリシー タブで **編集** を選択します。

5. ポリシーステートメントを既存のポリシーに追加します。
6. [Save changes] (変更の保存) をクリックします。

 Note

デフォルト AWS マネージドキー (aws/secretsmanager) を使用する場合、KMS キーポリシーを変更する必要はありません。

AWS 変換ジョブを設定する

1. AWS Transform ジョブで、Connect to resources に移動します。
2. Connect source code repository を選択します。
3. 認証方法として PAT Connector を選択します。
4. ステップ 2 のシークレット ARN を入力します。
5. (オプション) カスタマー管理の KMS キーを使用した場合は、KMS キー ARN を入力します。
6. リポジトリとブランチを選択します。
7. [続行] をクリックしてください。

AWS Transform は、シークレットへのアクセスに必要なアクセス許可を持つ IAM ロールを自動的に作成します。

トークンのローテーションとメンテナンス

有効期限が切れる前に PAT トークンをローテーションする責任があります。トークンをローテーションするには:

1. 同じアクセス許可でソースコードプロバイダーに新しい PAT を生成します。
2. のシークレット値を更新します AWS Secrets Manager。
3. Transform AWS ジョブが新しいトークンでリポジトリにアクセスできることを確認します。
4. ソースコードプロバイダーの古い PAT を取り消します。

PAT コネクタの問題のトラブルシューティング

アクセス拒否 — 無効な PAT

PAT の有効期限が切れていないことを確認します。PAT にプロバイダーに必要なスコープがあることを確認します。PAT が正しく保存されていることを確認します AWS Secrets Manager。

シークレットを取得できません

シークレット ARN が正しいことを確認します。ジョブログをチェックして、AWS Transform が IAM ロールを作成したことを確認します。カスタマー管理の KMS キーを使用している場合は、キーポリシーを確認します。

アクセス権限の不足

PAT には、オペレーションに必要なスコープがない可能性があります。必要なスコープで PAT を再生成し、シークレット値を更新します AWS Secrets Manager。

セットアップ AWS CodeConnections

AWS CodeConnections は、OAuth 2.0 認可フローを通じて一時的な OAuth 認証情報を自動的に取得するマネージドプロバイダー統合を使用します。アクセス許可はプロバイダーアプリで設定され、によって完全に管理されます AWS。アプリを一度承認すると、はすべての認証情報管理 AWS を処理します。

1. SQL Server モダナイゼーションジョブで、リソースに接続するに移動します。
2. Connect source code repository を選択します。
3. 既存の接続がない場合は、接続の作成を選択します。
4. リポジトリプロバイダーを選択します。
 - GitHub / GitHub Enterprise
 - GitLab.com
 - Bitbucket Cloud
 - Azure リポジトリ
5. プロバイダーの承認フローに従います。
6. 認可後、接続を選択します。

リポジトリとブランチを選択する

1. リストからリポジトリを選択します。
2. 変換するブランチ (通常はメイン、マスター、または開発) を選択します。
3. (オプション) .NET アプリケーションがリポジトリルートにない場合は、サブディレクトリを指定します。
4. [続行] をクリックしてください。

Note

AWS Transform は、変換されたコードの新しいブランチを作成します。通常のコードレビュープロセスを通じて、変更を確認してマージできます。

リポジトリアクセスの承認

GitHub およびその他のプラットフォームの場合、リポジトリ管理者は接続リクエストを承認する必要があります。

1. AWS Transform には検証リンクが表示されます。
2. このリンクをリポジトリ管理者と共有します。
3. 管理者は、リポジトリ設定でリクエストを確認して承認します。
4. 管理者がリクエストを承認すると、接続ステータスは「承認済み」に変わります。

Important

組織のポリシーによっては、承認プロセスに時間がかかる場合があります。変化に応じて計画を立てましょう。

ステップ 4: デプロイコネクタを作成する (オプション)

変換されたアプリケーションを AWS アカウントにデプロイする場合は、デプロイコネクタを選択するオプションがあります。

デプロイコネクタをセットアップする

1. アプリケーションをデプロイする場合は「はい」を選択します。No を選択すると、このステップはスキップされます。
2. 変換されたアプリケーションをデプロイする AWS アカウントを追加します。
3. コネクタを簡単に覚えやすい名前を追加する
4. コネクタリクエストを承認のために送信します。

デプロイコネクタの承認

AWS アカウント管理者は、デプロイコネクタの接続リクエストを承認する必要があります。

1. AWS 変換は検証リンクを表示します
2. このリンクを AWS アカウント管理者と共有する
3. 管理者は、リポジトリ設定でリクエストを確認して承認します。
4. 承認されると、接続ステータスは「承認済み」に変わります。

Important

組織のポリシーによっては、承認プロセスに時間がかかる場合があります。変化に応じて計画を立てましょう。

ステップ 5: リソースを確認する

データベースとリポジトリに接続すると、AWS Transform は必要なすべてのリソースにアクセスでき、変換の準備が整っていることを確認します。

AWS 変換の検証内容

- データベース接続: 接続がアクティブ、ユーザーが必要なアクセス許可を持っている、データベースにアクセスできる、バージョンがサポートされている
- リポジトリアクセス: リポジトリへのアクセス、ブランチの存在、.NET プロジェクトファイルの検出、データベース接続の検出
- 環境の準備状況: VPC 設定が DMS をサポート、必要な AWS サービスロールが存在する、ネットワーク接続が確立されている、リージョンの互換性が確認されている

プリフライトチェックリストを確認する

1. ジョブプランのリソースの確認に移動します。
2. チェックリスト項目を確認します。
 - # データベース接続の検証
 - # リポジトリへのアクセスが確認されました
 - # サポートされている .NET バージョン
 - # Entity Framework または ADO.NET が検出されました
 - # ネットワーク設定が有効
 - # 必要なアクセス許可の付与
3. すべての項目が完了と表示されたら、続行を選択します。
4. 警告やエラーが表示される項目がある場合は、先に進む前に対処してください。

ステップ 6: 検出と評価

AWS Transform は SQL Server データベースと .NET アプリケーションを分析して、モダナイゼーションの範囲と複雑さを理解します。

検出された内容

- データベースオブジェクト: テーブル、ビュー、インデックス、ストアドプロシージャ、関数、トリガー、制約、データ型、計算列、ID 列、外部キー関係
- アプリケーションコード: .NET プロジェクト構造、Entity Framework モデルと設定、ADO.NET データアクセスコード、データベース接続文字列、ストアドプロシージャ呼び出し、コード内の SQL クエリ
- 依存関係: どのアプリケーションがどのデータベースを使用するか、データベース間の依存関係、共有ストアドプロシージャ、一般的なデータアクセスパターン

検出プロセス

- AWS 変換はリソースの確認後に自動的に検出を開始します
- 検出には通常、データベースのサイズとアプリケーションの複雑さに応じて 5~15 分かかります。
- 作業ログの進行状況をモニタリングする
- AWS 変換は、オブジェクトが検出されるとリアルタイムの更新を表示します

検出結果を確認する

検出が完了したら、検出と評価に移動して確認します。

データベース分析:

- オブジェクト数: テーブル、ビュー、ストアドプロシージャ、関数、トリガーの数
- 複雑さスコア: 変換の複雑さの評価 (低、中、高)
- アクション項目: 人間の注意が必要なオブジェクト
- サポートされている機能: が自動的に変換されるデータベース機能
- サポートされていない機能: 回避策が必要な機能

アプリケーション分析:

- プロジェクトタイプ: ASP.NET Core、コンソールアプリ、クラスライブラリなど
- .NET バージョン: 検出された .NET Core バージョン
- データアクセスフレームワーク: Entity Framework バージョンまたは ADO.NET
- データベース接続: 見つかった接続文字列の数
- コードの複雑さ: 変換の複雑さの評価

依存関係マップ:

- application-to-database の関係の視覚的表現
- クロスデータベースの依存関係
- 共有コンポーネント

複雑さ評価について

AWS 変換はモダナイゼーションを 3 つのカテゴリに分類します。

複雑さ	特性	期待される成果
低 (クラス A)	標準 SQL パターン (ANSI SQL)、 シンプルなストアドプロシージャ、 基本データ型、標準設定のエンティティフレームワーク	予想される最小限の人的介入、高い自動化成功率

複雑さ	特性	期待される成果
中 (クラス B)	高度な T-SQL パターン、ビジネスロジックを使用した複雑なストアドプロシージャ、ユーザー定義関数、計算列	人間による介入が必要で、専門家によるレビューが推奨されます
高 (クラス C)	CLR アセンブリ、リンクサーバー、サービスブローカー、複雑な全文検索	重要な人的リファクタリングが必要、段階的アプローチを検討する

評価レポート

AWS Transform は、以下を含む詳細な評価レポートを生成します。

- 大まかな概要を含むエグゼクティブサマリー
- データベースインベントリを完了する
- アプリケーションインベントリ
- 変換準備状況の割合
- 労力の見積もり
- リスク評価と緩和戦略
- (推奨されるアプローチ)

評価レポートはオフラインレビュー用にダウンロードし、ステークホルダーと共有できます。

ステップ 7: ウェーブプランを生成して確認する

複数のデータベースとアプリケーションを持つ大規模な資産の場合、AWS Transform は論理グループでモダナイゼーションをシーケンスするウェーブプランを生成します。

ウェーブプランとは

ウェーブプランは、以下に基づいてモダナイゼーションをフェーズ (ウェーブ) に整理します。

- データベースとアプリケーション間の依存関係
- ビジネスの優先順位

- リスク許容度
- リソースの可用性
- 技術的な複雑さ

各ウェーブには、依存関係を損なうことなく一緒にモダナイズできるデータベースとアプリケーションのグループが含まれています。

ウェーブプランを確認する

1. ジョブプランで Wave 計画に移動する
2. 提案されたウェーブを確認する
3. ウェーブごとに、以下を確認してください。
 - 含まれるデータベース
 - 含まれるアプリケーション
 - 他のウェーブの依存関係
 - 推定変換時間
 - 複雑さレベル
 - デプロイ可能なアプリケーション

ウェーブプランをカスタマイズする

次の 2 つの方法で、ビジネスニーズに合わせてウェーブプランをカスタマイズできます。

JSON の使用:

1. すべてのウェーブのダウンロードを選択して、すべてのウェーブを含む JSON ファイルを取得します。
2. JSON のウェーブを変更するには、以下を実行します。
 - ウェーブ間のデータベースの移動
 - ウェーブを小さなグループに分割する
 - ウェーブをマージする
 - ウェーブシーケンスの変更
 - スコープからのデータベースの追加または削除
3. ウェーブプランのアップロードを選択して JSON ファイルをコンソールにアップロードします。

4. AWS 変換は変更を検証し、依存関係が侵害された場合に警告します
5. ウェーブの確認を選択してウェーブプランを更新します

チャットの使用:

エージェントとチャットし、リポジトリとデータベースを特定のウェーブに移動するように依頼することで、ウェーブプランを変更できます。このアプローチは、ウェーブに小さな編集を加える必要がある場合に適しています。

Important

ウェーブをカスタマイズするときに、依存関係が尊重されていることを確認します。データベースの前に依存アプリケーションを変換すると、問題が発生する可能性があります。

単一データベースのモダナイゼーション

単一のデータベースとアプリケーションをモダナイズする場合、AWS Transform は 1 つのウェーブでシンプルな計画を作成します。ウェーブプランニングなしで直接変換に進むことができます。

ウェーブプランを承認する

1. 確認してカスタマイズした後 (必要に応じて)、ウェーブプランの承認を選択します。
2. AWS 変換はウェーブプランをロックし、変換に進みます
3. ウェーブプランの編集を選択して、後でプランを変更できます。

ステップ 8: スキーマ変換

AWS Transform は、テーブル、ビュー、ストアドプロシージャ、関数、トリガーなど、SQL Server データベーススキーマを Aurora PostgreSQL に変換します。

スキーマ変換の仕組み

AWS Transform AWS は、生成 AI で拡張された DMS Schema Conversion を使用して以下を行います。

- SQL Server スキーマと関係を分析する
- SQL Server から PostgreSQL と同等のデータ型をマッピングする

- T-SQL を PL/pgSQL に変換する
- ID 列、計算列、制約を処理する
- 変換と参照整合性を検証する
- 人間によるレビューが必要なオブジェクトのアクション項目を生成する

サポートされている変換

自動変換:

- テーブル、ビュー、インデックス
- プライマリキーと外部キー
- 制約とデフォルト値を確認する
- 最も一般的なデータ型
- シンプルなストアードプロシージャ
- 基本的な関数とトリガー
- ID 列 (SERIAL または GENERATED に変換)
- ほとんどの計算列

人間によるレビューが必要になる場合があります。

- 高度な T-SQL を使用した複雑なストアードプロシージャ
- SQL Server 固有の関数 (GETUTCDATE、SUSER_SNAME など)
- 複雑な式を持つ計算列
- 全文検索インデックス
- XML データ型オペレーション
- HIERARCHYID データ型 (Itree 拡張機能が必要)

自動的に変換されません。

- CLR アセンブリ
- リンクサーバー
- サービスブローカー
- SQL Server エージェントジョブ

スキーマ変換を開始する

1. ジョブプランでスキーマ変換に移動する
2. 変換設定を確認します。
 - ターゲット PostgreSQL バージョン
 - 拡張機能オプション (ltree、PostGIS など)
 - 命名規則
3. 変換の開始を選択する
4. 作業ログの進行状況をモニタリングする
5. 変換には通常、データベースオブジェクトの数に応じて 10～30 分かかります。

変換結果を確認する

変換が完了したら、スキーマ変換の確認に移動します。

変換の概要:

- 変換されたオブジェクト: 正常に変換されたオブジェクトの数
- アクション項目: 人間による注意が必要なオブジェクト
- 警告: レビューすべき潜在的な問題
- エラー: 変換できなかったオブジェクト

オブジェクトタイプによるレビュー:

- テーブル: データ型マッピング、制約、インデックス
- ストアドプロシージャ: T-SQL から PL/pgSQL への変換
- 関数: 関数の署名とロジックの変更
- トリガー: トリガー構文とタイミングの変更

アクション項目を確認する

1. アクション項目の表示を選択する
2. アクション項目ごとに、以下を確認します。
 - オブジェクト名: データベースオブジェクト

- 問題タイプ: 注意が必要なもの
- 重要度: 重大、警告、または情報
- 推奨事項: 推奨される解決方法
- 元のコード: SQL Server バージョン
- 変換されたコード: PostgreSQL バージョン

3. アクション項目ごとに、次のことができます。

- 承認: 変換されたコードを使用する
- 変更: 変換されたコードを編集する
- 後の のフラグ: 変換後に人間によるレビュー用にマークする

例: ストアドプロシージャ変換

SQL Server 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
```

変換された PostgreSQL PL/pgSQL:

```
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;
```

変更点:

- TABLE を返す関数に変換されたプロシージャ
- p_ というプレフィックスが付いたパラメータ名
- TOP を LIMIT に変換
- EXTRACT に変換された DATEDIFF
- GETUTCDATE() を NOW() に変換
- 小文字に変換された列名 (PostgreSQL 規則)

スキーマ変換を承認する

1. すべてのアクション項目を確認し、必要な変更を加えた後
2. スキーマ変換の承認を選択する
3. AWS Transform は、Aurora PostgreSQL へのデプロイ用に変換されたスキーマを準備します

Note

変換されたスキーマは、オフラインレビューまたはバージョン管理用の SQL スクリプトとしてダウンロードできます。

ステップ 9: データ移行 (オプション)

AWS Transform には、SQL Server から Aurora PostgreSQL にデータを移行するためのオプションが用意されています。データ移行はオプションであり、スキーマとコード変換のみが必要な場合はスキップできます。

データ移行オプション

オプション 1: 本番稼働用データ移行

DMS AWS を使用して実際の本番稼働データを移行します。

- すべてのデータの完全な初期ロード
- テスト中の継続的レプリケーション (CDC)
- 最小ダウンタイムカットオーバー
- データの検証と整合性チェック

オプション 2: データ移行をスキップする

スキーマとコードのみを変換します。

- 開発/テスト環境に役立ちます
- データを個別に移行する場合
- proof-of-conceptプロジェクトの場合

データ移行を設定する

1. ジョブプランでデータ移行に移動する
2. 移行オプションを選択します。
 - 本番稼働用データの移行
 - データ移行をスキップする
3. 本番データを移行する場合は、以下を設定します。
 - 移行タイプ: 全ロード、または全ロード + CDC
 - 検証: データ検証を有効にする
 - パフォーマンス: DMS インスタンスサイズ

- 「> 移行の開始」を選択します。

本番稼働用データ移行プロセス

本番データを移行することを選択した場合:

1. 初期同期: AWS DMS はすべてのテーブルの全ロードを実行します
2. 継続的なレプリケーション: (CDC が有効になっている場合) データの同期を維持します
3. 検証: 行数とデータの整合性を検証します
4. カットオーバーの準備: 最終同期の準備

移行タイムライン:

- 小規模データベース (< 10 GB): 30 分 ~ 2 時間
- 中規模データベース (10 ~ 100 GB): 2 ~ 8 時間
- 大規模なデータベース (> 100 GB): 8 時間以上

データ検証

AWS 変換は、移行されたデータを次のチェックで検証します。

- 行数の比較 (ソースとターゲット)
- プライマリキーの整合性
- 外部キー関係
- データ型の互換性
- 計算された列の結果
- Null 値処理

ステップ 10: アプリケーションコード変換

AWS Transform は、.NET アプリケーションコードを SQL Server ではなく Aurora PostgreSQL と連携するように変換します。リポジトリ内のターゲットブランチ名に、変換されたソースコードをコミットするように要求します。ブランチ名を入力すると、AWS Transform は新しいブランチを作成し、PostgreSQL データベースと一致するように変換を開始します。

変換される内容

エンティティフレームワークの変更:

- データベースプロバイダー: UseSqlServer() → UseNpgsql()
- 接続文字列: SQL Server 形式 → PostgreSQL 形式
- データ型マッピング: SQL Server タイプ → PostgreSQL タイプ
- DbContext 設定: SQL Server 固有 → PostgreSQL 固有
- 移行ファイル: PostgreSQL との互換性のために更新

ADO.NET 変更:

- 接続クラス: SqlConnection → NpgsqlConnection
- コマンドクラス: SqlCommand → NpgsqlCommand
- データリーダー: SqlDataReader → NpgsqlDataReader
- パラメータ: SqlParameter → NpgsqlParameter
- SQL 構文: T-SQL → PostgreSQL SQL

設定の変更:

- appsettings.json の接続文字列
- データベースプロバイダー NuGet パッケージ
- 依存関係インジェクション設定
- Startup/Program.cs 設定

コード変換を開始する

1. ジョブプランのアプリケーション変換に移動する
2. 変換設定を確認します。
 - ターゲット .NET バージョン (アップグレードの場合)
 - PostgreSQL プロバイダーのバージョン
 - コードスタイルのプリファレンス
3. 変換の開始を選択する
4. 作業ログの進行状況をモニタリングする

5. 変換には通常、コードベースサイズに応じて 15～45 分かかります。

ステップ 11: 変換結果を確認する

デプロイに進む前に、完全な変換結果を確認して、すべてテストの準備が整っていることを確認します。

変換されたコードは、リポジトリブランチからダウンロードできます。

- ローカルテストと検証
- IDE でのコードレビュー
- CI/CD パイプラインとの統合
- バージョン管理コミット

変換の概要をダウンロードして、変換の一部として AWS 変換によって行われた自然言語の変更を確認することもできます。

変換の概要

1. ジョブプランで変換の概要に移動する
2. 全体的な結果を確認します。
 - スキーマ変換: 変換されたオブジェクト、アクション項目、警告
 - データ移行: テーブルの移行、行の転送、検証ステータス
 - コード変換: ファイルの変更、行の変更、問題の解決
 - 準備状況スコア: デプロイの全体的な準備状況

変換レポートの生成

AWS Transform は包括的な変換レポートを生成します。

1. レポートの生成を選択する
2. レポートタイプを選択します。
 - エグゼクティブサマリー: ステークホルダー向けの概要
 - 技術的な詳細: 完全な変換ドキュメント
 - アクション項目: 必要なヒューマンタスクのリスト
3. レポートのダウンロードを選択する

レポートには以下が含まれます。

- 変換の範囲と目的
- 変換されたオブジェクトとコード
- 発生した問題と解決策
- 検証結果
- デプロイ準備状況の評価
- テストに関する推奨事項

ステップ 12: 検証とテスト

本番環境にデプロイする前に、変換されたアプリケーションが Aurora PostgreSQL で正しく動作することを確認します。

検証タイプ

自動検証: AWS Transform は自動チェックを実行します。

- ソースデータベースに対するスキーマの検証
- データ整合性の検証
- クエリ同等性テスト
- 接続文字列の検証
- 設定の検証

人間による検証: 追加のテストを実行する必要があります。

- アプリケーション機能の機能テスト
- 他のシステムとの統合テスト
- パフォーマンステストとベンチマーク
- ユーザー受け入れテスト
- セキュリティテスト

自動検証を実行する

1. ジョブプランの検証に移動する

2. 検証の実行を選択する

3. AWS 変換は検証テストを実行します。

- データベース接続
- スキーマの互換性
- データ整合性
- アプリケーションビルド
- 基本的な機能

4. 検証結果を確認します。

- 合格: 成功したテスト
- 失敗: 注意が必要なテスト
- 警告: レビューすべき潜在的な問題

テストチェックリスト

データベース機能:

- アクセス可能なすべてのテーブル
- ストアドプロシージャが正しく実行される
- 関数は期待される結果を返す
- 適切に発射されるトリガー
- 適切に適用される制約
- インデックスによるクエリパフォーマンスの向上

アプリケーション機能:

- アプリケーションが正常に起動する
- データベース接続が確立されました
- CRUD オペレーションが正しく機能する
- ストアドプロシージャ呼び出しが成功
- トランザクションの適切なコミット/ロールバック
- エラー処理が期待どおりに機能する

データ整合性:

- 行カウントの一致ソース
- プライマリキー一意
- 外部キーが有効
- 計算された列が正しい
- Null 処理が適切
- 互換性のあるデータ型

パフォーマンス:

- クエリレスポンスの許容時間
- 接続プーリングの設定
- 最適化インデックス
- N+1 クエリの問題なし
- バッチオペレーションの効率化
- リソース使用率の妥当性

ステップ 13: デプロイ

検証が成功したら、モダナイズされたアプリケーションとデータベースを本番環境にデプロイします。

デプロイオプション

- Amazon ECS と Amazon EC2 Linux

デプロイ前チェックリスト

本番環境にデプロイする前に:

- すべての検証テストに合格しました
- パフォーマンステストが完了しました
- セキュリティレビューの完了
- バックアップおよびロールバック計画の文書化
- モニタリングとアラートの設定
- 新しい環境でトレーニングされたチーム

- デプロイの通知を受けたステークホルダー
- スケジュールされたメンテナンスウィンドウ

Amazon ECS へのデプロイ

1. ジョブプランのデプロイに移動する
2. ECS にデプロイを選択する
3. デプロイ設定を構成する:
 - クラスター: ECS クラスターを選択または作成する
 - サービス: ECS サービスを設定する
 - タスク定義: 生成されたタスク定義を確認する
 - ロードバランサー: ALB/NLB を設定する
 - 自動スケーリング: スケーリングポリシーを設定する
4. infrastructure-as-code を確認する (CloudFormation テンプレートまたは AWS CDK コード)
5. [デプロイ] をクリックします。

デプロイのモニタリング

AWS Transform はアプリケーションをデプロイします。

1. Aurora PostgreSQL クラスターを作成します
2. データベーススキーマを適用します
3. データをロードします (該当する場合)
4. アプリケーションコンテナをデプロイします
5. ロードバランサーを設定します
6. 自動スケーリングを設定する

デプロイの進行状況をモニタリングし、以下を確認します。

- インフラストラクチャーのプロビジョニング
- データベースの初期化
- アプリケーションのデプロイ
- ヘルスチェックの合格

- アクセス可能なアプリケーション
- データベース接続が機能する
- 通常のオペレーションを示すログ

デプロイ後の検証

デプロイ後:

煙テスト:

- 重要な機能を検証する
- キーユーザーワークフローをテストする
- 統合ポイントを確認する
- エラー率をモニタリングする

パフォーマンスモニタリング:

- 応答時間を追跡する
- データベースクエリのモニタリング
- リソース使用率を確認する
- アプリケーションログを確認する

ユーザー検証:

- ユーザー受け入れテストを実施する
- フィードバックの収集
- 問題に対処する
- 学んだ教訓を文書化する

ロールバック手順

デプロイ後に問題が発生した場合:

即時ロールバック:

- 以前のアプリケーションバージョンに戻す

- SQL Server に戻す (使用可能な場合)
- 必要に応じてバックアップから復元する

部分ロールバック:

- 特定のコンポーネントをロールバックする
- データベースの変更を保持する
- アプリケーションコードのみを元に戻す

フォワード修正:

- Aurora PostgreSQL バージョンにホットフィックスを適用する
- 更新されたアプリケーションコードをデプロイする
- 解決をモニタリングする

 Important

必要に応じてロールバックを有効にするために、カットオーバー後も SQL Server データベースを一定期間使用できるようにします。

デプロイ後の最適化

デプロイが成功した後:

パフォーマンスチューニング:

- スロークエリの最適化
- 接続プールの設定を調整する
- Aurora PostgreSQL パラメータをファインチューニングする
- インデックスの確認と最適化

コスト最適化:

- 適切なサイズの Aurora インスタンス
- 自動スケーリングを適切に設定する

- ストレージ設定を確認する
- バックアップ保持の最適化

モニタリングのセットアップ:

- CloudWatch ダッシュボードを設定する
- アラートの設定
- 拡張モニタリングを有効にする
- Performance Insights を設定する

ドキュメント:

- ランブックを更新する
- アーキテクチャの変更を文書化する
- 運用チームのトレーニング
- トラブルシューティングガイドを作成する

変換後のテストとデプロイ

変換されたアプリケーションをランディングゾーンにデプロイし、アプリケーションと変換されたデータベース間の接続を検証し、オプションで CI/CD パイプラインを設定します。サンプルの CI/CD パイプライン設定は、デプロイ自動化の実装に役立つリファレンスとして生成されます。この最終フェーズでは、モダナイゼーション全体を検証します。

リソースを設定する

IAM ロール、インスタンスプロファイル、S3 バケットなど、デプロイで使用するリソースを設定する必要があります。AWS Transform には、S3 バケットを作成したり、IAM に変更を加えたりするためのアクセス許可はありません。このアクションは、新しいロールを作成するアクセス許可を持つユーザーが実行する必要があります。通常はアカウント管理者です。

プロセスを簡素化するために、自動的に次の設定スクリプトが用意されています。

- 必要な CloudFormation テンプレートを実行します。
- ECS サービスにリンクされたロールを作成します
- 必要なすべての IAM リソースをセットアップします

⚠ Important

提供されたセットアップスクリプトをそのまま実行します。デプロイプロセスに影響する可能性があるため、テンプレートまたはスクリプトのリソース名を変更しないでください。

Linux の場合:

```
./setup.sh --region <YOUR-Region>
```

Windows の場合:

```
./setup.ps1 -Region <YOUR-Region>
```

VPC のセキュリティグループを確認する

AWS Transform は、アプリケーションを Aurora PostgreSQL インスタンスと同じサブネットおよび同じセキュリティグループにデプロイします。

アプリケーションが Aurora インスタンスに接続するには、このセキュリティグループに同じセキュリティグループへの接続を許可するアウトバウンドルールと、同じセキュリティグループからの接続を許可するインバウンドルールがあることを確認します。

アプリケーションを選択して設定する

AWS 変換 UI には、デプロイできるアプリケーションのリストが表示されます。アプリケーションを選択し、Configure and Commit をクリックして、以下を指定できる設定ウィンドウを開きます。

- アプリケーションのターゲットインフラストラクチャ: ECS または EC2 インスタンスのいずれかを選択する
- アプリケーションまたはインスタンスのパラメータ

設定ウィンドウを完了して閉じたら、AWS 変換します。

- 設定に基づいてデプロイアーティファクトを生成します
- 以下を含むこれらのアーティファクトをリポジトリにコミットします。
 - デプロイスクリプト
 - CloudFormation テンプレート

- CI/CD パイプライン設定のサンプル (デプロイ自動化を設定するためのリファレンスとして使用できます)

コミットが完了すると、アプリケーションのステータスは設定済みおよびコミット済みに変わります。その後、アプリケーションを選択し、右上のデプロイボタンを使用してデプロイを開始できます。

ECS デプロイ

ターゲット ECS を持つすべてのアプリケーションは、アカウントで作成された新しい ECS クラスターにデプロイされます。各アプリケーションは独自の専用コンテナにデプロイされます。コンテナごとに複数のアプリケーションはサポートされていません。必要な CPU やメモリなど、アプリケーションごとにコンテナパラメータを個別に設定します。

EC2 デプロイ

提案された EC2 インスタンスのいずれかを選択し、インスタンスタイプなどのパラメータを設定できます。複数のアプリケーションを同じ EC2 インスタンスにデプロイできます。

- 複数のアプリケーションを 1 つのインスタンスにデプロイするには、設定中にこれらのアプリケーションごとに同じターゲット EC2 インスタンスを選択します。
- ターゲットとして同じインスタンスが選択されているすべてのアプリケーションは、そのインスタンスにデプロイされ、異なるランダムポートが割り当てられます。
- あるアプリケーションのインスタンスパラメータを編集し、同じインスタンスをターゲットとする別のアプリケーションを編集すると、その変更はそのインスタンスにデプロイされたすべてのアプリケーションに影響することに注意してください。

デプロイの成功

デプロイが成功すると、アプリケーションは Aurora PostgreSQL で実行され、ライセンスコストの削減、パフォーマンスの向上、クラウドネイティブのスケラビリティが実現します。明示的に停止しない限り、ジョブは開いたままになります。これにより、更新されたデプロイパラメータを使用して再デプロイできるオプションが可能になります。

ベストプラクティスとトラブルシューティング

モダナイゼーションプロセスにおけるベストプラクティス、一般的な問題とその解決策。

ECS アプリケーションログ

CloudWatch ログ

- すべての ECS コンテナログは CloudWatch Logs に自動的に送信されます
- ロググループの CloudWatch コンソールでログにアクセスする
- ロググループの命名形式: /aws/ecs/{application-name}
- 各コンテナインスタンスは、グループ内に新しいログストリームを作成します。

ログの表示

AWS コンソール経由:

- CloudWatch > Log Groups に移動する
- アプリケーションのロググループを選択する
- 関連するログストリームを選択してコンテナログを表示する

CLI AWS の使用:

```
aws logs get-log-events --log-group-name /aws/ecs/your-app-name --log-stream-name your-stream-name
```

一般的なログの場所

- アプリケーションログ: CloudWatch Logs
- ECS サービスイベント: ECS コンソール > クラスター > サービス > イベントタブ
- コンテナのヘルス/ステータス: ECS コンソール > クラスター > サービス > タスクタブ

データベース接続管理

アプリケーションはデータベース接続設定に環境変数を使用します

接続の問題が発生した場合:

- 環境変数の現在の接続設定を確認する
- 必要に応じてデータベース接続文字列を変更するように環境変数を更新する

- 接続文字列の変更は、アプリケーションを再デプロイすることなく、環境変数の更新を通じて行うことができます。

データベース接続の問題

問題: AWS 変換を SQL Server に接続できない

解決方法:

- AWS Transform と SQL Server 間のネットワーク接続を検証する
- セキュリティグループのルールで適切なポートアクセスを確認する (1433)
- Secrets Manager でデータベース認証情報を確認する
- 作成したユーザーを使用してデータベースのアクセス許可をテストする
- SQL Server が混合モード認証用に設定されていることを確認します。
- シークレットに必要なタグがあることを確認する (プロジェクト: atx-db-modernization、所有者: database-connector)

ファイアウォールとセキュリティグループの問題

問題: 接続タイムアウトまたは「データベースに到達できません」エラー

根本原因: セキュリティグループまたはネットワーク ACLs ブロックする

解決方法:

1. セキュリティグループ設定の検証:

- SQL Server セキュリティグループに、DMS Schema Conversion セキュリティグループからのポート 1433 を許可するインバウンドルールがあることを確認します。
- ソースが IP アドレスではなくセキュリティグループ ID (sg-0123456789abcdef0 など) であることを確認します。
- DMS Schema Conversion セキュリティグループがインスタンスプロファイルで正しく指定されていることを確認します。
- 競合する拒否ルールがないことを確認します。

2. ネットワーク ACLs:

- サブネットレベルのネットワーク ACLs ポート 1433 でインバウンドトラフィックを許可することを確認する

- ネットワーク ACLs リターントラフィックのアウトバウンドエフェメラルポートを許可していることを確認する
- データベースサブネットと DMS サブネットネットワーク ACLs

3. VPC 設定の確認:

- DMS Schema Conversion インスタンスと SQL Server が同じ VPC にあるか、適切な VPC ピアリングがあることを確認します。
- ルートテーブルがサブネット間のトラフィックを許可することを確認する
- ファイアウォールアプライアンスがトラフィックをブロックしていないことを確認する

4. 接続のテスト:

- DMS Schema Conversion と同じサブネットでテスト EC2 インスタンスを起動する
- DMS Schema Conversion と同じセキュリティグループをアタッチする
- telnet または SQL Server Management Studio を使用して SQL Server への接続をテストする
- テストが成功した場合、問題は AWS 変換設定です。失敗した場合、問題はネットワーク/ファイアウォールです。

よくある間違い: ポート 1433 から 0.0.0.0/0 (すべてのソース) を開くことはセキュリティ上のリスクです。セキュリティグループベースのアクセスコントロールを常に使用して、DMS Schema Conversion セキュリティグループのみにアクセスを制限します。

スキーマ変換の問題

問題: スキーマ変換で多くのアクション項目が表示される

解決方法:

- 変換レポートのアクション項目を確認する
- 影響に基づいて優先順位を付ける
- 複雑な SQL 変換に Amazon Q Developer を使用する
- ガイダンスについては AWS 、 サポートにお問い合わせください。
- 複雑なデータベースの段階的アプローチを検討する

アプリケーション変換の問題

問題: アプリケーション変換の構築に失敗する

解決方法:

- 変換レポートのビルドエラーを確認する
- 必要に応じてプライベート NuGet フィードを設定する
- 必要に応じてパッケージリファレンスを更新する
- Windows 固有の依存関係を確認する
- 変換ログで詳細なエラーを確認する

データ移行の問題

問題: データ移行の検証が失敗する

解決方法:

- 特定の障害の検証レポートを確認する
- データ型マッピングを確認する
- ID 列の設定を確認する (GENERATED BY DEFAULT と GENERATED ALWAYS)
- 計算された列式を確認する
- 複雑なデータ問題については、AWS サポートにお問い合わせください。

リソースクリーンアップの問題

問題: 変換ジョブがリソースエラーで失敗する

解決方法:

- 既存の DMS リソース (移行プロジェクト、データプロバイダー、インスタンスプロファイル) を確認する
- 失敗したリソースまたは不完全なリソースを以前の試行からクリーンアップする
- シークレットの削除がスケジュールされていないことを確認する
- DMS と Aurora PostgreSQL のサービスクォータを確認する
- クリーンアップで問題が解決しない場合は、AWS サポートにお問い合わせください。

デプロイに関する問題

問題: 変換されたアプリケーションが Aurora PostgreSQL に接続できない

解決方法:

- PostgreSQL の接続文字列形式を確認する
- セキュリティグループのルールを確認する
- Secrets Manager でデータベース認証情報を検証する
- SSL/TLS が正しく設定されていることを確認します。
- psql または pgAdmin を使用して接続をテストする

追加のヘルプの取得

AWS サポートに連絡するときは、以下を提供してください。

- 変換ジョブ ID
- AWS アカウント ID
- リージョン
- エラーメッセージとスクリーンショット
- 変換ログ (AWS 変換コンソールで使用可能)

AWS 変換におけるセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、AWS とお客様の間の責任共有です。[責任共有モデル](#)ではこれをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、で AWS サービスを実行するインフラストラクチャを保護する責任を担います AWS クラウド。は、お客様が安全に使用できるサービス AWS も提供します。サードパーティーの監査者は、[AWS コンプライアンスプログラム](#)コンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。AWS 変換に適用されるコンプライアンスプログラムの詳細については、「[コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム](#)」を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、AWS Transform を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティとコンプライアンスの目的を達成するために AWS 変換を設定する方法を示します。また、AWS Transform リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

トピック

- [でのデータ保護 AWS Transform](#)
- [AWS Transform の Identity and Access Management](#)
- [AWS 変換のコンプライアンス検証](#)
- [AWS 変換の耐障害性](#)
- [AWS Transform およびインターフェイスエンドポイント \(AWS PrivateLink\)](#)
- [VPC から AWS Transform ウェブアプリケーションにアクセスする](#)

でのデータ保護 AWS Transform

責任 AWS [共有モデル](#)は、でのデータ保護に適用されます AWS Transform。このモデルで説明されているように、AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。ユーザーは、このインフラストラクチャでホストされるコンテンツに対する管理を維持する責任があります。また、AWS のサービス 使用する のセキュリティ設定および管理タスクについても責任を負います。データプライバシーの詳細については、「[データプライバシーのよくある質問](#)」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログ に投稿された「[AWS 責任共有モデルおよび GDPR](#)」ブログを参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント、AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要なアクセス許可のみを各ユーザーに付与できます。また、次の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 以降が推奨されます。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- などの高度なマネージドセキュリティサービスを使用して Amazon Macie、に保存されている機密データの検出と保護を支援します Amazon S3。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-2 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「[連邦情報処理規格 \(FIPS\) 140-2](#)」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報を、[タグ](#)、または名前フィールドなどの自由形式のテキストフィールドに含めないことを強くお勧めします。これは、AWS Transform、API AWS Management Console、AWS Command Line Interface (AWS CLI)、または SDK AWS のサービス を使用して または他の を操作する場合も同様です。AWS SDKs タグ、または名前に使用される自由記述のテキストフィールドに入力したデータは、請求または診断ログに使用される場合があります。がコンテンツ AWS Transform を使用方法の詳細については、「」を参照してください [AWS Transform サービスの改善](#)。

AWS Transform は、質問、そのレスポンス、コンソールメタデータやコードなどの追加のコンテキストを IDE に保存して、質問に対するレスポンスを生成します。データの暗号化方法については、

「[でのデータの暗号化 AWS Transform](#)」を参照してください。AWS が質問したいいくつかの質問をどのように使用するか AWS Transform、およびその回答を使用してサービスを改善する方法については、「」を参照してください[AWS Transform サービスの改善](#)。

では AWS Transform、データは AWS Transform プロファイルが作成された AWS リージョンに保存されます。

クロス推論を使用すると、へのリクエストは、データが保存されている地域内の別のリージョンで処理 AWS Transform される場合があります。詳細については、「[クロスリージョン推論](#)」を参照してください。

トピック

- [でのクロスリージョン処理 AWS Transform](#)
- [でのデータの暗号化 AWS Transform](#)
- [AWS Transform サービスの改善](#)

でのクロスリージョン処理 AWS Transform

以下のセクションでは、クロスリージョン推論とクロスリージョン呼び出しを使用して AWS Transform サービスを提供する方法について説明します。

クロスリージョン推論

AWS Transform は Amazon Bedrock を利用しており、クロスリージョン推論を使用してさまざまなトラフィックを分散 AWS リージョン し、大規模言語モデル (LLM) 推論のパフォーマンスと信頼性を向上させます。クロスリージョン推論を使用すると、以下が可能になります。

- 高需要期間中のスループットと耐障害性の向上
- パフォーマンスの向上
- Amazon Bedrock でホストされている最も強力な LLMs に依存する新しく起動された AWS Transform 機能へのアクセス

クロスリージョン推論リクエストは、データが最初に存在する地域の一部 AWS リージョン である内に保持されます。たとえば、米国の AWS Transform 設定から行われたリクエストは、米国の AWS リージョン 内に保持されます。クロスリージョン推論はデータの保存場所を変更しませんが、リクエストと出力結果はデータが存在するリージョン外に移動する可能性があります。すべてのデー

タは、Amazon の安全なネットワークを介した送信中に暗号化されます。クロスリージョン推論の使用に追加コストはかかりません。

クロスリージョン推論は、データの保存先には影響しません。の使用時にデータを保存する場所については AWS Transform、[「」を参照してください](#) [でのデータ保護 AWS Transform](#)。

AWS Transform クロスリージョン推論でサポートされているリージョン

に対して行う特定のリクエストでは、クロスリージョン呼び出しが必要になる AWS Transform 場合があります。次の表は、リクエストが発生した地域に応じて、リクエストをルーティングできるリージョンを示しています。

送信元リージョン	送信先リージョン
米国東部 (バージニア北部) (us-east-1)	米国東部 (バージニア北部) (us-east-1)
	米国東部 (オハイオ) (us-east-2)
	米国西部 (オレゴン) (us-west-2)
ヨーロッパ (フランクフルト) (eu-central-1)	ヨーロッパ (フランクフルト) (eu-central-1)
	欧州 (ストックホルム) (eu-north-1)
	欧州 (ミラノ) (eu-south-1)
	欧州 (スペイン) (eu-south-2)
	欧州 (アイルランド) (eu-west-1)
アジアパシフィック (ムンバイ) (ap-south-1)	欧州 (パリ) (eu-west-3)
	アジアパシフィック (東京) (ap-northeast-1)
	アジアパシフィック (ソウル) (ap-northeast-2)
	アジアパシフィック (大阪) (ap-northeast-3)
	アジアパシフィック (ムンバイ) (ap-south-1)
	アジアパシフィック (ハイデラバード) (ap-south-2)

送信元リージョン	送信先リージョン
	アジアパシフィック (シンガポール) (ap-south-east-1) アジアパシフィック (シドニー) (ap-south-east-2) アジアパシフィック (メルボルン) (ap-south-east-4)
アジアパシフィック (東京) (ap-northeast-1)	アジアパシフィック (東京) (ap-northeast-1) アジアパシフィック (大阪) (ap-northeast-3)
アジアパシフィック (ソウル) (ap-northeast-2)	アジアパシフィック (東京) (ap-northeast-1) アジアパシフィック (ソウル) (ap-northeast-2) アジアパシフィック (大阪) (ap-northeast-3) アジアパシフィック (ムンバイ) (ap-south-1) アジアパシフィック (ハイデラバード) (ap-south-2) アジアパシフィック (シンガポール) (ap-south-east-1) アジアパシフィック (シドニー) (ap-south-east-2) アジアパシフィック (メルボルン) (ap-south-east-4)
アジアパシフィック (シドニー) (ap-south-east-2)	アジアパシフィック (シドニー) (ap-south-east-2) アジアパシフィック (メルボルン) (ap-south-east-4)

送信元リージョン	送信先リージョン
ヨーロッパ (ロンドン) (eu-west-2)	ヨーロッパ (フランクフルト) (eu-central-1)
	欧州 (ストックホルム) (eu-north-1)
	欧州 (ミラノ) (eu-south-1)
	欧州 (スペイン) (eu-south-2)
	欧州 (アイルランド) (eu-west-1)
	ヨーロッパ (ロンドン) (eu-west-2)
	欧州 (パリ) (eu-west-3)
カナダ (中部) (ca-central-1)	商用 AWS リージョン + カナダ (中部) (ca-central-1)
南米 (サンパウロ) (sa-east-1)	商用 AWS リージョン + 南米 (サンパウロ) (sa-east-1)

使用できるリージョンの完全なリストについては AWS Transform、「」を参照してください [でサポートされているリージョン AWS Transform](#)。

クロスリージョンの知識

AWS Transform サービス、変換ワークフロー、または関連 AWS ドキュメントについて一般的な質問をすると、米国リージョンの場合は米国東部 (バージニア) (us-east-1)、他のすべてのリージョンの場合は欧州 (フランクフルト) (eu-central-1) にクロスリージョンリクエストを行い、ドキュメントを取得してリクエストを満たす AWS Transform ことができます。例えば、Lambda などの他の AWS のサービスの使用方法について質問する場合、AWS Transform クロスリージョン呼び出しを行って、質問に回答するための関連 AWS ドキュメントを取得できます。次の表は、リクエストが発生した地域に応じて、リクエストをルーティングできるリージョンを示しています。

送信元リージョン	送信先リージョン
米国東部 (バージニア北部) (us-east-1)	米国東部 (バージニア北部) (us-east-1)

送信元リージョン	送信先リージョン
ヨーロッパ (フランクフルト) (eu-central-1)	ヨーロッパ (フランクフルト) (eu-central-1)
ヨーロッパ (ロンドン) (eu-west-2)	ヨーロッパ (フランクフルト) (eu-central-1)
アジアパシフィック (東京) (ap-northeast-1)	ヨーロッパ (フランクフルト) (eu-central-1)
アジアパシフィック (シドニー) (ap-south-east-2)	ヨーロッパ (フランクフルト) (eu-central-1)
アジアパシフィック (ソウル) (ap-northeast-2)	ヨーロッパ (フランクフルト) (eu-central-1)
アジアパシフィック (ムンバイ) (ap-south-1)	ヨーロッパ (フランクフルト) (eu-central-1)
カナダ (中部) (ca-central-1)	ヨーロッパ (フランクフルト) (eu-central-1)
南米 (サンパウロ) (sa-east-1)	ヨーロッパ (フランクフルト) (eu-central-1)

この設定はデフォルトで有効になっています。アカウント管理者は、この設定を変更できます。この機能を無効にすると、が他のリージョンから知識を取得 AWS Transform する必要がある機能にアクセスできなくなります。これにより、応答の精度が低下する可能性があります。

によって行われたクロスリージョン呼び出しを無効にするには AWS Transform:

1. 初めてセットアップするときは AWS Transform、開始方法ページに移動し、設定を完了します。既存の AWS Transform 設定の場合は、「設定」ページに移動します。
2. 切り替え クロスリージョン呼び出しを有効にして AWS、関連する一般的な質問にオフ位置に回答します。

AWS Transform for Windows クロスリージョン推論

次の表は、推論プロファイルを呼び出すことができる送信元リージョンと、リクエストをルーティングできる送信先リージョンを示しています。

送信元リージョン	推論送信先リージョン
米国東部 (バージニア北部) (us-east-1)	米国東部 (バージニア北部) (us-east-1)

送信元リージョン	推論送信先リージョン
	米国東部 (オハイオ) (us-east-2)
	米国西部 (オレゴン) (us-west-2)
ヨーロッパ (フランクフルト) (eu-central-1)	ヨーロッパ (フランクフルト) (eu-central-1)
	欧州 (ストックホルム) (eu-north-1)
	欧州 (ミラノ) (eu-south-1)
	欧州 (スペイン) (eu-south-2)
	欧州 (アイルランド) (eu-west-1)
	欧州 (パリ) (eu-west-3)
アジアパシフィック (東京) (ap-northeast-1)	アジアパシフィック (東京) (ap-northeast-1)
	アジアパシフィック (大阪) (ap-northeast-3)
アジアパシフィック (シドニー) (ap-south-east-2)	アジアパシフィック (シドニー) (ap-south-east-2)
	アジアパシフィック (メルボルン) (ap-south-east-4)
アジアパシフィック (ソウル) (ap-northeast-2)	すべての商用リージョン
アジアパシフィック (ムンバイ) (ap-south-1)	すべての商用リージョン
カナダ (中部) (ca-central-1)	すべての商用リージョン

AWS Transform 移行のクロスリージョン推論

次の表は、推論プロファイルを呼び出すことができる送信元リージョンと、リクエストをルーティングできる送信先リージョンを示しています。

送信元リージョン	推論送信先リージョン
米国東部 (バージニア北部) (us-east-1)	すべての商用リージョン
ヨーロッパ (フランクフルト) (eu-central-1)	すべての商用リージョン
ヨーロッパ (ロンドン) (eu-west-2)	すべての商用リージョン
アジアパシフィック (東京) (ap-northeast-1)	アジアパシフィック (東京) (ap-northeast-1) アジアパシフィック (大阪) (ap-northeast-3)
アジアパシフィック (シドニー) (ap-south-east-2)	アジアパシフィック (シドニー) (ap-south-east-2) アジアパシフィック (メルボルン) (ap-south-east-4)
アジアパシフィック (ソウル) (ap-northeast-2)	すべての商用リージョン
アジアパシフィック (ムンバイ) (ap-south-1)	すべての商用リージョン
カナダ (中部) (ca-central-1)	すべての商用リージョン

AWS Transform メインフレームクロスリージョン推論用

AWS Transform メインフレームのは、Amazon Bedrock の地理的クロスリージョン推論を使用します。つまり、一部の呼び出しはソースリージョンの AWS リージョン 外部にルーティングされる可能性があります。次の表は、推論プロファイルを呼び出すことができる送信元リージョンと、リクエストをルーティングできる送信先リージョンを示しています。

送信元リージョン	推論送信先リージョン
米国東部 (バージニア北部) (us-east-1)	米国東部 (バージニア北部) (us-east-1) 米国東部 (オハイオ) (us-east-2) 米国西部 (オレゴン) (us-west-2)
ヨーロッパ (フランクフルト) (eu-central-1)	ヨーロッパ (フランクフルト) (eu-central-1)

送信元リージョン	推論送信先リージョン
	欧州 (ストックホルム) (eu-north-1) 欧州 (ミラノ) (eu-south-1) 欧州 (スペイン) (eu-south-2) 欧州 (アイルランド) (eu-west-1) 欧州 (パリ) (eu-west-3)
ヨーロッパ (ロンドン) (eu-west-2)	ヨーロッパ (フランクフルト) (eu-central-1) 欧州 (ストックホルム) (eu-north-1) 欧州 (ミラノ) (eu-south-1) 欧州 (スペイン) (eu-south-2) 欧州 (アイルランド) (eu-west-1) ヨーロッパ (ロンドン) (eu-west-2) 欧州 (パリ) (eu-west-3)
アジアパシフィック (東京) (ap-northeast-1)	アジアパシフィック (東京) (ap-northeast-1) アジアパシフィック (大阪) (ap-northeast-3)
アジアパシフィック (シドニー) (ap-south-east-2)	アジアパシフィック (シドニー) (ap-south-east-2) アジアパシフィック (メルボルン) (ap-south-east-4)
アジアパシフィック (ソウル) (ap-northeast-2)	すべての商用リージョン
アジアパシフィック (ムンバイ) (ap-south-1)	すべての商用リージョン

送信元リージョン	推論送信先リージョン
カナダ (中部) (ca-central-1)	カナダ (中部) (ca-central-1)
	米国東部 (バージニア北部) (us-east-1)
	米国東部 (オハイオ) (us-east-2)
	米国西部 (オレゴン) (us-west-2)
南米 (サンパウロ) (sa-east-1)	すべての商用リージョン

AWS Transform カスタム

AWS Transform カスタムは、Amazon Bedrock の地理的クロスリージョン推論を使用します。つまり、一部の呼び出しはソースリージョンの AWS リージョン 外部にルーティングされる可能性があります。次の表は、推論プロファイルを呼び出すことができる送信元リージョンと、リクエストをルーティングできる送信先リージョンを示しています。

送信元リージョン	推論送信先リージョン
米国東部 (バージニア北部) (us-east-1)	米国東部 (バージニア北部) (us-east-1)
	米国東部 (オハイオ) (us-east-2)
	米国西部 (オレゴン) (us-west-2)
ヨーロッパ (フランクフルト) (eu-central-1)	ヨーロッパ (フランクフルト) (eu-central-1)
	欧州 (ストックホルム) (eu-north-1)
	欧州 (ミラノ) (eu-south-1)
	欧州 (スペイン) (eu-south-2)
	欧州 (アイルランド) (eu-west-1)
	欧州 (パリ) (eu-west-3)
ヨーロッパ (ロンドン) (eu-west-2)	ヨーロッパ (フランクフルト) (eu-central-1)

送信元リージョン	推論送信先リージョン
	欧州 (ストックホルム) (eu-north-1)
	欧州 (ミラノ) (eu-south-1)
	欧州 (スペイン) (eu-south-2)
	欧州 (アイルランド) (eu-west-1)
	ヨーロッパ (ロンドン) (eu-west-2)
	欧州 (パリ) (eu-west-3)
アジアパシフィック (東京) (ap-northeast-1)	すべての商用リージョン
アジアパシフィック (シドニー) (ap-south-east-2)	アジアパシフィック (シドニー) (ap-south-east-2)
	アジアパシフィック (メルボルン) (ap-south-east-4)
アジアパシフィック (ソウル) (ap-northeast-2)	すべての商用リージョン
アジアパシフィック (ムンバイ) (ap-south-1)	すべての商用リージョン
カナダ (中部) (ca-central-1)	カナダ (中部) (ca-central-1)
	米国東部 (バージニア北部) (us-east-1)
	米国東部 (オハイオ) (us-east-2)
	米国西部 (オレゴン) (us-west-2)

でのデータの暗号化 AWS Transform

このトピックでは、転送中の暗号化と保管中の暗号化 AWS Transform について固有の情報を提供します。

AWS Transform はデフォルトで暗号化を提供し、AWS 所有キーを使用した暗号化で保管中の機密データを保護します。

暗号化タイプ

転送中の暗号化

顧客間 AWS Transform、および AWS Transform とダウンストリームの依存関係間の通信は、TLS 1.2 以降の接続を使用して保護されます。

Important

検出アカウントを接続すると、AWS Transform はそのアカウントでユーザーに代わって Amazon S3 バケットを作成します。そのバケットはデフォルトでは SecureTransport 有効になっていません。バケットポリシーに安全なトランスポートを含める場合は、ポリシーを更新する必要があります。詳細については、[Amazon S3 のセキュリティのベストプラクティス](#)」を参照してください。

保管中の暗号化

AWS Transform は、Amazon DynamoDB と Amazon Simple Storage Service (Amazon S3) を使用して保管中のデータを保存します。保管中のデータは、デフォルトで AWS 暗号化ソリューションを使用して暗号化されます。は、AWS Key Management Service (AWS KMS) の AWS 所有キーを使用した暗号化でデータを AWS Transform 暗号化します。データを暗号化する AWS マネージドキーを保護するためのアクションを実行する必要はありません。詳細については、「AWS Key Management Service デベロッパーガイド」の「[AWS 所有キー](#)」を参照してください。

データ型	AWS 所有のキー暗号化	カスタマー マネージドキーの暗号化 (オプション)
カスタマー バケット データ	有効	有効
Amazon S3 バケットに保存されているコードやドキュメントなどの顧客の入出力		
アーティファクトストア	有効	有効
S3 バケットに保存されているコード変換の一環としての中間アーティファクト		

データ型	AWS所有のキー暗号化	カスタマーマネージドキーの暗号化 (オプション)
ジョブの目標 Amazon S3 バケットに保存されているジョブに対する顧客のインテント	有効	有効
チャットメッセージ Amazon S3 バケットに保存 AWS Transform されている顧客と の間のメッセージ Amazon S3	有効	有効
チャットナレッジベース Amazon OpenSearch に保存され、AWS Bedrock を介して処理される AWS Transform およびカスタマーチャットに関連するインデックス化されたデータ	有効	有効

注: お客様は、上記のすべてのデータ型の暗号化に使用する独自のカスタマーマネージドキー (CMK) を登録できます。

カスタマーマネージドキーは、KMS キーへのアクセスを制御することでデータへのアクセスを直接制御するために作成、所有、管理する AWS アカウントの KMS キーです。対称キーのみがサポートされます。独自の KMS キーの作成については、[「デベロッパーガイド」の「キーの作成」](#)を参照してください。AWS Key Management Service

カスタマーマネージドキーを使用する場合、AWS Transform は KMS 許可を使用し、承認されたユーザー、ロール、またはアプリケーションが KMS キーを使用できるようにします。AWS Transform 管理者が設定中に暗号化にカスタマーマネージドキーを使用することを選択すると、権限が作成されます。この権限は、エンドユーザーが保管中のデータ暗号化に暗号化キーを使用できるようにするものです。権限の詳細については、「[Grants in AWS KMS](#)」を参照してください。

AWS 所有キー (デフォルト)

Note

AWS 所有キー — デフォルトでこれらのキー AWS Transform を使用して、個人を特定できるデータを自動的に暗号化します。AWS 所有キーを表示、管理、使用したり、その使用を監査したりすることはできません。ただし、データを暗号化するキーを保護するためのアクションの実施やプログラムの変更を行う必要はありません。詳細については、「Key Management Service デベロッパーガイド」の「AWS 所有 AWSキー」を参照してください。

デフォルトでは、保管中のデータを暗号化することで、機密データの保護に伴う運用のオーバーヘッドと複雑な作業を軽減できます。同時に、セキュリティを重視したアプリケーションを構築することで、暗号化のコンプライアンスと規制の厳格な要件を満たすことができます。

この暗号化レイヤーを無効にしたり、別の暗号化タイプを選択したりすることはできませんが、変換の作成時にカスターマネージドキーを選択することで、既存の AWS 所有暗号化キーに 2 番目の暗号化レイヤーを追加できます。

カスターマネージドキー (オプション)

カスターマネージドキー — 作成、所有、管理する対称カスターマネージドキーの使用 AWS Transform をサポートし、AWS 所有キーを使用して既存の暗号化に 2 番目の暗号化レイヤーを追加します。この暗号化層はユーザーが完全に制御できるため、次のようなタスクを実行できます。

- キーポリシーの策定と維持
- IAM ポリシーとグラントの策定と維持
- キーポリシーの有効化と無効化
- キー暗号化マテリアルのローテーション
- タグを追加する
- キーエイリアスの作成
- キー暗号化マテリアルのローテーション
- タグを追加する
- キーエイリアスの作成
- 削除のためのキースケジューリング

- 詳細については、「AWS Key Management Service デベロッパーガイド」の「[カスタマーマネージドキー](#)」を参照してください。

Note

AWS Transform は、AWS 所有キーを使用した保管時の暗号化を自動的に有効にし、個人を特定できるデータを無償で保護します。ただし、カスタマーマネージドキーの使用には AWS KMS 料金が適用されます。料金の詳細については、「[AWS Key Management Service 料金](#)」を参照してください。

詳細については AWS KMS、[「とは」を参照してください AWS Key Management Service](#)。

で 許可 AWS Transform を使用する方法 AWS Key Management Service

AWS Transform には、カスタマーマネージドキーを使用するための許可が必要です。

カスタマーマネージドキーで暗号化された [リソース名] を作成すると、AWS Transform は CreateGrant リクエストを送信してユーザーに代わって許可を作成します AWS Key Management Service。の権限 AWS Key Management Service は、カスタマーアカウントの KMS キー AWS Transform へのアクセスを許可するために使用されます。

AWS Transform では、次の内部オペレーションにカスタマーマネージドキーを使用する権限が必要です。

- [KMS API] リクエストを に送信 AWS KMS して、〔リソース名〕の作成時に入力された対称カスタマーマネージド KMS キー ID が有効であることを確認します。
- [KMS API] リクエストを に送信 AWS KMS して、カスタマーマネージドキーによって暗号化されたデータキーを生成します。
- [KMS API] リクエストを AWS KMS に送信して、暗号化されたデータキーを復号し、データの暗号化に使用できます。

グラントへのアクセスの取り消しや、カスタマーマネージドキーに対するサービスのアクセスの取り消しは、いつでもできます。その場合、カスタマーマネージドキーによって暗号化されたデータにはアクセス AWS Transform できず、そのデータに依存するオペレーションに影響します。たとえば、アクセス AWS Transform できない暗号化された [リソース名] から [リソース名] を取得しようとする、オペレーションは AccessDeniedException エラーを返します。

カスタマーマネージドキーを作成する

対称カスタマーマネージドキーは、AWS マネジメントコンソールまたは AWS Key Management Service APIs を使用して作成できます。

対称カスタマーマネージドキーを作成するには、AWS Key Management Service デベロッパーガイドの「[対称暗号化 KMS キーの作成](#)」のステップに従います。

AWS Transform リソースでカスタマーマネージドキーを使用するには、キーポリシーで次の API オペレーションを許可する必要があります。

kms:CreateGrant – カスタマーマネージドキーに権限を追加します。指定された KMS キーへのアクセスを制御する権限を付与します。これにより、AWS Transform 必要な権限付与オペレーションへのアクセスが可能になります。詳細については、「AWS Key Management Service デベロッパーガイド」の「[AWS KMS でのグラント](#)」を参照してください。

これにより、AWS Transform は以下を実行できます。

- [KMS API ([GenerateDataKeyWithoutPlainText/GenerateDatakey](#))] を呼び出して暗号化されたデータキーを生成し、保存します。これは、データキーがすぐに暗号化に使用されるわけではないためです。
- [KMS API ([Decrypt](#))] を呼び出して、保存された暗号化されたデータキーを使用して暗号化されたデータにアクセスします。
- サービスに [RetireGrant](#) を許可するには、リタイアプリンシパルを設定します。
- [kms:DescribeKey](#) – [サービス名] がキーを検証できるようにするカスタマーマネージドキーの詳細を提供します。

SQL モダライゼーションの暗号化

[SQL Server のモダライゼーション](#) プロセス中に、EC2 インスタンス、ECS クラスター、ECR イメージ、S3 オブジェクトなどのリソースがアカウントに AWS Transform 作成されます。アカウント内のデータを暗号化するためのカスタマーマネージドキーを提供するオプションがあります。KMS キーに CreatedFor キーと値 AWSTransform でタグ付けする必要があります。このポリシーの例では、デフォルトのキーポリシーがない場合に、キーポリシーに必要な最小限のアクセス許可を一覧表示します。デフォルトのキーポリシーがある場合でも、ECS がキーを使用できるように、デフォルトのキーポリシーに加えて KMS キーを手動で更新する必要があります。

```
"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"
    }
  }
}

```

```
    }
  }
}
]
```

データベース移行の暗号化

データベース移行プロセス中に、は AWS Database Migration Service (DMS) AWS Transform を使用してデータベースデータを移行します。オプションでカスタマーマネージドキーを指定して、移行中にデータベースデータを暗号化できます。

次のポリシーには 3 つのステートメントが含まれています。最初のステートメントは AWS KMS、キーのアクセス許可をルートアカウントに付与します。2 番目のステートメントでは、コネクタロールが検出中にキーを検証できるようにします。3 番目のステートメントでは、コネクタロールが DMS オペレーションに キーを使用することを許可します。

```
"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"
  }
}
}
```

カスタマーマネージド KMS キーの使用

カスタマーマネージド KMS キーを作成した後、AWS Transform 管理者は AWS Transform コンソールでキーを指定して、それを使用してデータを暗号化する必要があります。

でデータを暗号化するようにカスタマーマネージドキーを設定するには AWS Transform、管理者が使用するためのアクセス許可が必要です AWS KMS。

カスタマーマネージドキーで暗号化された機能を使用するには、AWS Transform がカスタマーマネージドキーにアクセスできるようにするアクセス許可が必要です。

の使用中に KMS 許可に関連するエラーが表示された場合は AWS Transform、許可を更新して AWS Transform が許可を作成できるようにする必要があります。必要なアクセス許可を自動的に設定するには、AWS Transform コンソールに移動し、ページ上部のバナーでアクセス許可の更新を選択します。が AWS Transform 許可を作成できるようにするには、AWS Transform コンソールページのアクセス許可を更新する必要があります。

カスタマーマネージドキー AWS Transform へのアクセスを許可する

次のポリシーステートメントの例では、キーへのアクセスを許可することで、カスタマーマネージドキーで暗号化された機能 AWS Transform にアクセスするためのアクセス許可をユーザーに付与します。このポリシーは、管理者が暗号化用にカスタマーマネージドキーをセットアップ AWS Transform している場合に使用する必要があります。

Note

この情報は、CMK を使用した SQL Server モダナイゼーションデプロイワークフローには適用されません。

```
"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 サービスの改善

AWS Transform 最も関連性の高い情報を提供するため、サービスの改善のために AWS Transform、お客様からの質問 AWS Transform や回答など、 の特定のコンテンツを使用する場合があります。このページでは、AWS がどのようなコンテンツを使用するのか、またオプトアウトの方法について説明します。

AWS Transform サービスの改善に使用されるコンテンツ

のサービス改善 AWS Transform のために、 の特定のコンテンツを使用する場合があります。は、一般的な質問へのより良い回答の提供、運用上の問題の修正 AWS Transform 、デバッグ、モデルトレーニングなどのために、このコンテンツを使用する AWS Transform 場合があります。

がサービスの改善に使用する AWS 可能性のあるコンテンツには、 への質問 AWS Transform や、 が AWS Transform 生成するレスポンスとコードが含まれます。

オプトアウトの方法

サービス改善のためのコンテンツ AWS Transform の使用をオプトアウトする方法は、使用する環境によって異なります AWS Transform。

で AI サービスのオプトアウトポリシー AWS Transform を設定する場合 AWS Organizations。詳細については、「AWS Organizations ユーザーガイド」の「[AI サービスオプトアウトポリシー](#)」を参照してください。

IDE Visual Studio でのコンテンツの共有をオプトアウトするには、次の手順を使用します。

[ツール] -> [オプション] -> [AWS Toolkit] -> [Amazon Q] に移動します。

[Share Amazon Q Content with AWS] を [True] または [False] に切り替えます。

AWS Toolkit for でのテレメトリデータの共有をオプトアウトするには Visual Studio、次の手順を使用します。

1. [ツール] で [オプション] を選択します。
2. [オプション] ペインで、[AWS Toolkit] を選択し、[全般] を選択します。
3. AWS ツールキットによる使用状況情報の収集を許可するの選択を解除します。

AWS Transform の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つです。IAM 管理者は、誰を認証 (サインイン) し、誰に AWS Transform リソースの使用を許可する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

トピック

- [オーディエンス](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)
- [AWS Transform と IAM の連携方法](#)
- [AWS Transform のアイデンティティベースのポリシーの例](#)
- [トラブルシューティング アイデンティティとアクセスの AWS 変換](#)
- [のサービスにリンクされたロールの使用 AWS Transform](#)
- [AWS AWS Transform の マネージドポリシー](#)
- [AWS 変換アクセス許可リファレンス](#)

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします ([「トラブルシューティング アイデンティティとアクセスの AWS 変換」](#) を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します ([「AWS Transform と IAM の連携方法」](#) を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します ([「AWS Transform のアイデンティティベースのポリシーの例」](#) を参照)

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

AWS IAM アイデンティティセンター (IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーテッド ID としてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の [「AWS アカウントにサインインする方法」](#) を参照してください。

プログラムによるアクセスの場合、は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の [「API リクエストに対するAWS 署名バージョン 4」](#) を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント ルートユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザー認証情報を必要とするタスクについては、「IAM ユーザーガイド」の [「ルートユーザー認証情報が必要なタスク」](#) を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーテッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID Directory Service ソースの認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「[IAM アイデンティティセンターとは](#)」を参照してください。

IAM ユーザーとグループ

[IAM ユーザー](#)は、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用することをお勧めします。詳細については、IAM ユーザーガイドの「[ID プロバイダーとのフェデレーションを使用して にアクセスする必要がある AWS](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「[IAM ユーザーに関するユースケース](#)」を参照してください。

IAM ロール

[IAM ロール](#)は、特定のアクセス許可を持つアイデンティティであり、一時的な認証情報を提供します。ユーザーから [IAM ロール \(コンソール\) に切り替えるか、または API オペレーションを呼び出すことで、ロール](#)を引き受けることができます。AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「[ロールを引き受けるための各種方法](#)」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイドの [IAM でのクロスアカウントリソースアクセス](#) を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられたときにアクセス許可を定義します。は、プリンシパルがリクエストを行うときにこれらのポリシー AWS を評価します。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの詳細については、「IAM ユーザーガイド」の「[JSON ポリシー概要](#)」を参照してください。

管理者は、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

アイデンティティベースのポリシー

アイデンティティベースのポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイドの [カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する](#) を参照してください。

アイデンティティベースのポリシーは、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーとインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「[管理ポリシーとインラインポリシーのいずれかを選択する](#)」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の上限を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可境界](#)」を参照してください。

- サービスコントロールポリシー (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「[サービスコントロールポリシー](#)」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「[リソースコントロールポリシー \(RCP\)](#)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。詳細については、「IAM ユーザーガイド」の「[セッションポリシー](#)」を参照してください。

複数のポリシータイプ

1つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「[ポリシー評価ロジック](#)」を参照してください。

AWS Transform と IAM の連携方法

IAM を使用して AWS Transform へのアクセスを管理する前に、AWS Transform で使用できる IAM 機能を確認してください。

IAM 機能	AWS 変換のサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	部分的

IAM 機能	AWS 変換のサポート
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	あり
サービスリンクロール	はい

AWS Transform およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要については、「IAM ユーザーガイド」の[AWS「IAM と連携する のサービス」](#)を参照してください。

AWS Transform のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「[カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する](#)」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「[IAM JSON ポリシーの要素のリファレンス](#)」を参照してください。

AWS Transform のアイデンティティベースのポリシーの例

アイデンティティベースのポリシーの AWS 変換の例を表示するには、「」を参照してください [AWS Transform のアイデンティティベースのポリシーの例](#)。

AWS Transform 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があ

げられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーテッドユーザー、またはを含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。詳細については、IAM ユーザーガイドの[IAM でのクロスアカウントリソースアクセス](#)を参照してください。

AWS 変換のポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

AWS 変換アクションのリストを確認するには、「サービス認可リファレンス」の[AWS 「変換のアクション、リソース、および条件キー」](#)を参照してください。

カスタム AWS 変換のアクションのリストを確認するには、「サービス認可リファレンス」の[「カスタム AWS 変換のアクション、リソース、および条件キー」](#)を参照してください。

AWS Transform のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
transform
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [  
    "transform:action1",  
    "transform:action2"  
]
```


アイデンティティベースのポリシーの AWS 変換の例を表示するには、「」を参照してください [AWS Transform のアイデンティティベースのポリシーの例](#)。

AWS 変換のポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、[Amazon リソースネーム \(ARN\)](#) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"

```

AWS Transform リソースタイプとその ARNs 「サービス認可リファレンス」の [AWS 「Transform のアクション、リソース、および条件キー」](#) を参照してください。

リソースタイプとその ARNs 「サービス認可リファレンス」の [AWS 「変換カスタムのアクション、リソース、および条件キー」](#) を参照してください。AWS

アイデンティティベースのポリシーの AWS 変換の例を表示するには、「」を参照してください [AWS Transform のアイデンティティベースのポリシーの例](#)。

AWS 変換のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの [条件演算子](#) を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を

一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」の[AWS「グローバル条件コンテキストキー」](#)を参照してください。

AWS 変換条件キーのリスト、および条件キーを使用できるアクションとリソースを確認するには、「サービス認可リファレンス」の[AWS「変換のアクション、リソース、および条件キー」](#)を参照してください。

条件キーのリストと、AWS 変換カスタムの条件キーを使用できるアクションとリソースを確認するには、「サービス認可リファレンス」の[AWS「変換カスタムのアクション、リソース、および条件キー」](#)を参照してください。

アイデンティティベースのポリシーの AWS 変換の例を表示するには、「」を参照してください[AWS Transform のアイデンティティベースのポリシーの例](#)。

AWS 変換ACLs

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

AWS 変換による ABAC

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの[条件要素](#)でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「[ABAC 認可でアクセス許可を定義する](#)」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「[属性ベースのアクセス制御 \(ABAC\) を使用する](#)」を参照してください。

AWS Transform での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、AWS リソースへの短期的なアクセスを提供し、フェデレーションまたはスィッチロールの使用時に自動的に作成されます。AWS では、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM ユーザーガイド」の「[IAM の一時的な認証情報](#)」および「[AWS のサービスと IAM との連携](#)」を参照してください。

AWS Transform のクロスサービスプリンシパルアクセス許可

転送アクセスセッション (FAS) のサポート: あり

転送アクセスセッション (FAS) は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウストリームサービス AWS のサービス へのリクエストをリクエストする を使用します。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

AWS 変換のサービスロール

サービスロールのサポート: あり

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの [AWS のサービスに許可を委任するロールを作成する](#)を参照してください。

Warning

サービスロールのアクセス許可を変更すると、AWS 変換機能が破損する可能性があります。AWS Transform が指示する場合にのみ、サービスロールを編集します。

AWS Transform のサービスにリンクされたロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービ

スにリンクされたロールは、に表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、AWS「」を参照してください。[のサービスにリンクされたロールの使用 AWS Transform](#)。

AWS Transform のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーとロールには AWS 変換リソースを作成または変更するアクセス許可はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーを作成する \(コンソール\)](#)」を参照してください。

各リソースタイプの ARN の形式など、AWS Transform で定義されるアクションとリソースタイプの詳細については、「サービス認可リファレンス」の「[AWS Transform のアクション、リソース、および条件キー](#)」を参照してください。ARNs

トピック

- [ポリシーに関するベストプラクティス](#)
- [AWS 変換コンソールの使用](#)
- [自分の権限の表示をユーザーに許可する](#)
- [管理者が AWS Transform を使用してアカウントからのコネクタリクエストを受け入れることを許可する](#)
- [管理者が既存の IAM Identity Center ユーザーを割り当て、新しい IAM Identity Center ユーザーを作成して AWS Transform に割り当てることを許可する](#)
- [管理者が AWS 変換を有効にすることを許可する](#)
- [IAM 認証情報を使用して AWS Transform へのアクセスをユーザーに許可する](#)

ポリシーに関するベストプラクティス

ID ベースのポリシーは、アカウント内の AWS 変換リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションでは、AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらはで使えます AWS アカウント。ユースケースに固有のAWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイドの [AWS マネージドポリシー](#) または [ジョブ機能のAWS マネージドポリシー](#) を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイドの [IAM でのポリシーとアクセス許可](#) を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する – ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションがなどの特定のを通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイドの [IAM JSON ポリシー要素:条件](#) を参照してください。
- IAM アクセスアナライザーを使用して IAM ポリシーを検証し、安全で機能的な権限を確保する – IAM アクセスアナライザーは、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイドの [IAM Access Analyzer でポリシーを検証する](#) を参照してください。
- 多要素認証 (MFA) を要求する – IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイドの [MFA を使用した安全な API アクセス](#) を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイドの [IAM でのセキュリティのベストプラクティス](#) を参照してください。

AWS 変換コンソールの使用

AWS 変換コンソールにアクセスするには、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、の AWS 変換リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリ

シーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

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

```

    }
  ]
}

```

管理者が AWS Transform を使用してアカウントからのコネクタリクエストを受け入れることを許可する

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-*"
    }
  ]
}

```

管理者が既存の IAM Identity Center ユーザーを割り当て、新しい IAM Identity Center ユーザーを作成して AWS Transform に割り当てることを許可する

次のポリシーは、管理者が AWS Transform コンソールで IAM Identity Center ユーザーとグループを管理するために必要なアクセス許可を付与します。このポリシーを使用すると、管理者はユーザーとグループの割り当てと削除、表示名の表示、新しい IAM Identity Center ユーザーの作成、アプリケーションの割り当ての管理、アプリケーションセッションの設定を行うことができます。

```

{
  "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"
      }
    }
  }
}

```

```

    }
  }
}
]
}

```

管理者が AWS 変換を有効にすることを許可する

次のポリシーは、管理者が AWS コンソールを介して AWS 変換を有効化および管理するために必要なアクセス許可を付与します。これには、プロファイルの管理、エージェントの設定、コネクタの管理が含まれます。このポリシーは管理者のみを対象としており、アカウントの信頼できるプリンシパルに限定する必要があります。

```

{
  "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": "*"
    }
  ]
}

```

IAM 認証情報を使用して AWS Transform へのアクセスをユーザーに許可する

次のポリシーは、IAM 認証情報を使用してウェブアプリケーションと APIs の AWS 変換へのアクセス権を IAM プリンシパルに付与します。このポリシーは、IAM のみのアクセス、および ID プロバイダーとともに有効になっている IAM アクセスに必要です。

region、*account-id*、を自分の値*profile-id*に置き換えます。プロファイル ID は、AWS 変換コンソールの設定で確認できます。

```

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

```

```
{
  "Sid": "AllowTransformAccess",
  "Effect": "Allow",
  "Action": "transform:AccessTransformProfile",
  "Resource": "arn:aws:transform:region:account-id:profile/profile-id"
}
```

アカウント内のすべてのプロファイルへのアクセスを許可するには、プロファイル ID にワイルドカードを使用します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTransformAccessAllProfiles",
      "Effect": "Allow",
      "Action": "transform:AccessTransformProfile",
      "Resource": "arn:aws:transform:region:account-id:profile/*"
    }
  ]
}
```

Note

transform:AccessTransformProfile アクションは、AWS 変換へのアクセスのみを許可します。ワークスペース内のアクションは、AWS 変換ワークスペースロール (管理者、コントリビューター、承認者、読み取り専用) によって制御され、IAM ポリシーによって制御されません。Workspace ロールは、ジョブの作成、共同作業者の管理、タスクの承認など、ユーザーができることを決定します。

トラブルシューティング アイデンティティとアクセスの AWS 変換

次の情報は、AWS Transform と IAM の使用時に発生する可能性がある一般的な問題の診断と修正に役立ちます。

トピック

- [Transform でアクションを実行する権限がありません AWS](#)
- [iam:PassRole を実行する権限がありません](#)
- [自分の 以外のユーザーに AWS Transform リソース AWS アカウント へのアクセスを許可したい](#)

Transform でアクションを実行する権限がありません AWS

アクションを実行する権限がないというエラーが表示された場合は、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要な AWS Transform:*GetWidget* アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: AWS
Transform:GetWidget on resource: my-example-widget
```

この場合、AWS Transform:*GetWidget* アクションを使用して *my-example-widget* リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

iam:PassRole を実行する権限がありません

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して、AWS Transform にロールを渡すことができるようにする必要があります。

一部の AWS のサービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡すアクセス許可が必要です。

次の例のエラーは、という IAM ユーザーがコンソールを使用して AWS 変換でアクションを実行marymajorしようとするると発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与されたアクセス許可が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに `iam:PassRole` アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 以外のユーザーに AWS Transform リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- AWS Transform がこれらの機能をサポートしているかどうかを確認するには、「」を参照してください [AWS Transform と IAM の連携方法](#)。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、「[IAM ユーザーガイド](#)」の「[所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する](#)」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「[サードパーティー AWS アカウント が所有する へのアクセスを提供する](#)」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの [外部で認証されたユーザー \(ID フェデレーション\) へのアクセスの許可](#) を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「[IAM でのクロスアカウントのリソースへのアクセス](#)」を参照してください。

のサービスにリンクされたロールの使用 AWS Transform

AWS Transform は AWS Identity and Access Management (IAM) [サービスにリンクされたロール](#) を使用します。サービスにリンクされたロールは、直接リンクされた一意のタイプの IAM ロールです AWS Transform。サービスにリンクされたロールは によって事前定義 AWS Transform されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

のサービスにリンクされたロールの使用 AWS Transform

AWS Transform は AWS Identity and Access Management (IAM) [サービスにリンクされたロール](#)を使用します。サービスにリンクされたロールは、直接リンクされた一意のタイプの IAM ロールです AWS Transform。サービスにリンクされたロールは によって事前定義 AWS Transform されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、 の設定 AWS Transform が簡単になります。 は、サービスにリンクされたロールのアクセス許可 AWS Transform を定義し、特に定義されている場合を除き、 のみがそのロールを引き受け AWS Transform ることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。これにより、AWS Transform リソースへのアクセス許可が誤って削除されないため、リソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスの詳細については、[AWS 「IAM と連携するサービス」](#)を参照し、「サービスにリンクされたロール」列で「はい」があるサービスを探します。サービスリンクロールに関するドキュメントをサービスで表示するには、リンクで [はい] を選択します。

のサービスにリンクされたロールのアクセス許可 AWS Transform

AWS Transform は、[AWSServiceRoleForAWSTransform](#) という名前のサービスにリンクされたロールを使用します。このサービスにリンクされたロールは、AWS Transform に使用状況情報を提供する機能を提供します。

AWSServiceRoleForAWSTransform サービスにリンクされたロールは、次のサービスを信頼してロールを引き受けます。

- `transform.amazonaws.com`

AWSServiceRoleForAWSTransform という名前のロールアクセス許可ポリシーにより AWS Transform、 は指定されたリソースに対して次のアクションを実行できます。

- `cloudwatch:PutMetricData`
 - AWS Transform オペレーションのためにカスタムメトリクスを CloudWatch に送信する

- 変換の進行状況、成功率、パフォーマンスメトリクスを追跡する
- 変換ワークフローのモニタリングとアラートを有効にする
- sso:DescribeApplication
 - Identity Center で特定のアプリケーションの詳細を表示する
 - 名前、説明、ステータス、設定などのアプリケーションメタデータを取得する
- sso:GetApplicationAssignmentConfiguration
 - アプリケーションの割り当て設定を取得する
 - ユーザー/グループをアプリケーションに割り当てるように設定する方法を確認する
- sso:ListApplicationAssignmentsForPrincipal
 - 特定のユーザーまたはグループに割り当てられたすべてのアプリケーションを一覧表示する (プリンシパル)
 - 特定の ID がアクセスできるアプリケーションを表示する
- IAM Identity Center 経由でアクセスすると、KMS で暗号化されたデータの復号を有効にします。暗号化コンテキストに有効な IAM Identity Center インスタンス ARN が含まれており、IAM Identity Center サービスエンドポイントを介してアクセスする必要がある場合にのみ機能します。
- Identity Store からアクセスしたときに KMS で暗号化されたデータの復号を許可します。暗号化コンテキストに有効な Identity Store ARN が含まれており、Identity Store サービスエンドポイントを介してアクセスする必要がある場合にのみ機能します。
- secretsmanager:GetSecretValue
 - 外部 ID プロバイダーのクライアントシークレットを保存するために使用される AWS Transform サービスにリンクされたシークレットにアクセスする
 - リソース: arn:aws:secretsmanager:*:*:secret:transform-preprod!*
 - 条件: transform-preprod サービスが所有し、同じアカウントからアクセスする必要があります
- support:CreateCase、support:DescribeCases、support:DescribeCommunications、support:AddCommunication
 - AWS Transform ウェブアプリケーションからプレミアムサポートケースを作成および管理します。
 - ケースの詳細とコミュニケーションを表示する
 - コミュニケーションを追加し、サポートケースを解決する

ユーザー、グループ、またはロールにサービスリンクロールの作成、編集、または削除を許可するには、アクセス許可を設定する必要があります。詳細についてはIAM ユーザーガイドの「[サービスにリンクされた役割のアクセス許可](#)」を参照してください。

のサービスにリンクされたロールの作成 AWS Transform

サービスリンクロールを手動で作成する必要はありません。AWS Transform で のプロファイルを作成すると AWS Management Console、 によってサービスにリンクされたロールが自動的に AWS Transform 作成されます。

このサービスリンクロールを削除した後で再度作成する必要がある場合は同じ方法でアカウントにロールを再作成できます。設定を更新すると、AWS Transform によってサービスにリンクされたロールが再度作成されます。

IAM コンソールまたは CLI AWS を使用して、サービス名でtransform.amazonaws.comサービスにリンクされたロールを作成することもできます。詳細については、IAM ユーザーガイドの「[サービスリンクロールの作成](#)」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

のサービスにリンクされたロールの編集 AWS Transform

AWS Transform では、AWSServiceRoleForAWSTransform サービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「[サービスリンクロールの編集](#)」を参照してください。

のサービスにリンクされたロールの削除 AWS Transform

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

Note

リソースを削除しようとしたときに AWS Transform サービスがロールを使用している場合は、削除が失敗する可能性があります。その場合は、数分待ってからオペレーションを再試行してください。

IAM を使用してサービスリンクロールを手動で削除するには

IAM コンソール、AWS CLI、または AWS API を使用して、AWSServiceRoleForAWSTransform サービスにリンクされたロールを削除します。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの削除](#)」を参照してください。

AWS Transform サービスにリンクされたロールでサポートされているリージョン

AWS Transform は、サービスが利用可能なすべてのリージョンでサービスにリンクされたロールの使用をサポートしているわけではありません。AWSServiceRoleForAWSTransform ロールは、次のリージョンで使用できます。詳細については、「[AWS リージョンとエンドポイント](#)」を参照してください。

リージョン名	リージョン識別子	でのサポート AWS Transform
米国東部 (バージニア北部)	us-east-1	はい
欧州 (フランクフルト)	eu-central-1	はい

AWS Transform Custom のサービスにリンクされたロールの使用

AWS Transform Custom は、AWSServiceRoleForAWSTransformCustom という名前のサービスにリンクされたロールを使用して、ユーザーに代わってメトリクスとログをアカウントに発行します。これらのメトリクスを使用すると、変換数、レイテンシー、ステータスコードをダッシュボードで直接モニタリングできます。ログは、変換中に発生した問題を可視化します。

この[サービスにリンクされたロール](#)は AWS Transform Custom によって事前定義されており、サービスに必要なアクセス許可のみが含まれています。アクセス許可を手動で追加する必要はなく、ロールは AWS Transform Custom によってのみ引き受けることができます。サービスにリンクされたロールの一般的な情報については、IAM [ユーザーガイドの「サービスにリンクされたロールの使用」](#)を参照してください。

AWS Transform Custom のサービスにリンクされたロールのアクセス許可

AWSServiceRoleForAWSTransformCustom サービスにリンクされたロールは、次のサービスを信頼してロールを引き受けます。

- `transform-custom.amazonaws.com`

AWSServiceRoleForAWSTransformCustom という名前のロールアクセス許可ポリシーにより、AWS Transform Custom は指定されたリソースに対して次のアクションを実行できます。

- `cloudwatch:PutMetricData` すべての AWS リソースで
 - オペレーションメトリクスを `AWS/TransformCustom` 名前空間の に発行する
 - 変換数、レイテンシー、ステータスコードを追跡する
 - `cloudwatch:namespace` 条件キーを介して `AWS/TransformCustom` 名前空間にスコープ設定
- `logs:CreateLogGroup` /`aws/TransformCustom` ロググループの `logs:PutRetentionPolicy` および
 - 変換ログを発行するための CloudWatch Logs ロググループを作成する
 - ロググループにログ保持ポリシーを設定する
 - アカウント内でのみアクセスを保証する `arn:aws:logs:*:*:log-group:/aws/TransformCustomaws:ResourceAccount` 条件で にスコープ
- `logs:CreateLogStream` `logs:PutLogEvents` ロググループ内の /`aws/TransformCustom` ログストリームの および
 - 変換オペレーションのログストリームを作成し、ログイベントを発行する
 - アカウント内でのみアクセスを保証する `aws:ResourceAccount` 条件 `arn:aws:logs:*:*:log-group:/aws/TransformCustom:log-stream:*` を持つ の範囲

ユーザー、グループ、またはロールにサービスリンクロールの作成、編集、または削除を許可するには、アクセス許可を設定する必要があります。詳細については IAM ユーザーガイド の「[サービスにリンクされた役割のアクセス許可](#)」を参照してください。

AWS Transform Custom のサービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。AWS Transform Custom CLI を使用して変換を実行すると、AWS Transform Custom によってサービスにリンクされたロールが自動的に作成されます。

このサービスリンクロールを削除した後で再度作成する必要がある場合は同じ方法でアカウントにロールを再作成できます。AWS Transform Custom CLI を使用して変換を実行すると、AWS Transform Custom によってサービスにリンクされたロールが再度作成されます。

IAM コンソールを使用して、AWSServiceRoleForAWSTransformCustom ユースケースでサービスにリンクされたロールを作成することもできます。AWS CLI または AWS API で、サービス名を使用

して `transform-custom.amazonaws.com` サービスにリンクされたロールを作成します。詳細については、IAM ユーザーガイドの「[サービスリンクロールの作成](#)」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

AWS Transform Custom のサービスにリンクされたロールの編集

AWS Transform Custom では、`AWSServiceRoleForAWSTransformCustom` サービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「[サービスリンクロールの編集](#)」を参照してください。

AWS Transform Custom のサービスにリンクされたロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。

`AWSServiceRoleForAWSTransformCustom` サービスにリンクされたロールは、アカウントに永続的リソースを作成しません。リソースをクリーンアップせずに直接削除できます。

このロールを削除し、後で AWS 変換カスタム CLI を使用して変換を実行すると、サービスにリンクされたロールが自動的に再作成されます。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、AWS CLI、または AWS API を使用し

て、`AWSServiceRoleForAWSTransformCustom` サービスにリンクされたロールを削除します。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの削除](#)」を参照してください。

AWS Transform Custom サービスにリンクされたロールでサポートされているリージョン

AWS Transform Custom は、AWS Transform Custom が利用可能なすべてのリージョンで、サービスにリンクされたロールの使用をサポートしています。サポートされているリージョンのリストについては、「[サポートされているリージョン](#)」を参照してください。

AWS AWS Transform の マネージドポリシー

AWS 管理ポリシーは、によって作成および管理されるスタンドアロンポリシーです AWS。AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有の[カスタマー管理ポリシー](#)を定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使えるようになったときに、AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「[AWS マネージドポリシー](#)」を参照してください。

AWS AWS 管理ポリシーの更新を変換する

2021 年 3 月 1 日以降の AWS Transform の AWS マネージドポリシーの更新に関する詳細を表示します。

変更	説明	日付
AWSTransformInfrastructureExecutorAccessBatch – 新しいポリシー	ソースコードのアップロード、結果の取得、AWS バッチジョブのステータスのモニタリング、ログの読み取り、変換スケジュールの管理など、Batch を使用して AWS Transform Continuous Modernization の評価と変換を実行するために必要なアクセス許可を付与する新しい AWS マネージドポリシーを追加しました。	2026 年 7 月 20 日
AWSTransformInfrastructureExecutorAccessEC2 – 新しいポリシー	ソースコードのアップロード、結果の取得、ジョブステータスのモニタリング、変換スケジュールの管理など、Amazon EC2 を使用して AWS Transform Continuous	2026 年 7 月 20 日

変更	説明	日付
	Modernization の評価と変換を実行するために必要なアクセス許可を付与する新しい AWS マネージドポリシーを追加しました。	
AWSTransformSecurityAgentExecutorAccess – 新しいポリシー	スキャンアーティファクトのアップロードや結果の取得など、自動化されたコードセキュリティレビューと修復のために AWS セキュリティエージェントサービスを呼び出すために必要なアクセス許可を AWS Transform Continuous Modernization に付与する新しい AWS マネージドポリシーを追加しました。	2026 年 6 月 30 日
AWSServiceRoleForAWSTransformCustom – 更新されたポリシー	AWS Transform custom がアカウントのログ/aws/TransformCustom グループにログを発行できるようにする CloudWatch Logs アクセス許可を追加しました。	2026 年 5 月 5 日
AWSTransformCustomExecuteTransformations – 更新されたポリシー	カスタムサービスにリンクされたロール (AWSServiceRoleForAWSTransformCustom) を AWS 変換して CloudWatch メトリクスをお客様のアカウントに放出できるようにするアクセス許可を追加しました。	2026 年 4 月 27 日

変更	説明	日付
AWSTransformCustomManageTransformations – 更新されたポリシー	カスタムサービスにリンクされたロール (AWSServiceRoleForAWSTransformCustom) を AWS 変換して CloudWatch メトリクスをお客様のアカウントに放出できるようにするアクセス許可を追加しました。	2026 年 4 月 27 日
AWSTransformCustomFullAccess – 更新されたポリシー	カスタムサービスにリンクされたロール (AWSServiceRoleForAWSTransformCustom) を AWS 変換して CloudWatch メトリクスをお客様のアカウントに放出できるようにするアクセス許可を追加しました。	2026 年 4 月 7 日
AWSServiceRoleForAWSTransformCustom – 新しいポリシー	カスタムサービスにリンクされたロールを AWS 変換するための新しい AWS 管理ポリシーを追加しました。このポリシーにより、AWS Transform custom は CloudWatch メトリクスをアカウントに発行できます。	2026 年 3 月 23 日
DBModProvisioningAndMigration – 新しいポリシー	このポリシーは、データベースのプロビジョニングと移行機能を付与します。	2026 年 3 月 24 日

変更	説明	日付
DBModDiscoveryAndAssessment – 新しいポリシー	包括的なデータベースモダライゼーションの検出および評価機能を提供する新しい AWS マネージドポリシーを追加しました。	2026 年 3 月 24 日
AWSTransformCustomFullAccess – 新しいポリシー	AWS Transform custom へのフルアクセスを提供する新しい AWS マネージドポリシーを追加しました。	2025 年 12 月 5 日
AWSTransformCustomExecuteTransformations – 新しいポリシー	AWS Transform custom で変換を実行するためのアクセスを提供する新しい AWS マネージドポリシーを追加しました。	2025 年 12 月 5 日
AWSTransformCustomManageTransformations – 新しいポリシー	AWS 変換カスタムで変換リソースを作成、更新、読み取り、削除したり、変換を実行したりするためのアクセスを提供する新しい AWS マネージドポリシーを追加しました。	2025 年 12 月 5 日

変更	説明	日付
AWSServiceRoleForAWSTransform – 更新されたポリシー	外部 ID プロバイダーのクライアントシークレットを保存するために使用される AWS 変換サービスにリンクされたシークレットにアクセスするためのアクセス許可を追加しました。 AWS Transform ウェブアプリからプレミアムサポートケースを作成するアクセス許可を追加しました。	2025 年 12 月 1 日
AWSTransformApplicationECSDeploymentPolicy – 更新されたポリシー	ECR 暗号化サポートのための IAM ロール検査アクセス許可、ECS サービスにリンクされたロールの作成、および KMS アクセス許可を追加しました。	2025 年 11 月 22 日
AWSTransformApplicationDeploymentPolicy – 更新されたポリシー	EC2 ネットワークアクセス許可、IAM ロール検査アクセス許可、S3 バケットリストアクセス許可、および拡張デプロイ機能の KMS 暗号化サポートを追加しました。	2025 年 11 月 22 日
AWSServiceRoleForAWSTransform – 更新されたポリシー	IAM Identity Center でカスタマーマネージドキーのサポートが追加されました。	2025 年 9 月 17 日

変更	説明	日付
AWSTransformApplicationDeploymentPolicy – 新しいポリシー	Amazon EC2 インスタンス、CloudFormation スタック、および関連リソースを作成および管理することで、AWS 変換された .NET アプリケーションを Transform がデプロイできるようにする新しい AWS マネージドポリシーを追加しました。	2025 年 8 月 28 日
AWSServiceRoleForAWSTransform – 更新されたポリシー	新しい ポリシーを追加しました。	2025 年 5 月 15 日

AWS マネージドポリシー: AWSServiceRoleForAWSTransform

このポリシーは、[AWSServiceRoleForAWSTransform](#) サービスリンクロール (SLR) にアタッチされます。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSServiceRoleForAWSTransform](#)」を参照してください。

AWS マネージドポリシー: AWSServiceRoleForAWSTransformCustom

このポリシーは、[AWSServiceRoleForAWSTransformCustom](#) サービスリンクロール (SLR) にアタッチされます。このロールにより、AWS Transform custom はユーザーに代わって CloudWatch メトリクスとログをアカウントに発行できます。

説明

このポリシーには、以下のアクセス許可が含まれています。

- Amazon CloudWatch – AWS/TransformCustom 名前空間で CloudWatch へのメトリクスの発行を許可します。これにより、CloudWatch ダッシュボードの変換数、レイテンシー、ステータスコードをモニタリングできます。

- Amazon CloudWatch Logs – ロググループの作成、ログストリーム、保持ポリシーの設定、ロググループへの /aws/TransformCustom ログイベントの発行を許可します。これにより、変換中に発生した問題を可視化できます。

AWS マネージドポリシー: AWSTransformApplicationDeploymentPolicy

このポリシーにより、AWS Transform は Amazon EC2 インスタンス、CloudFormation スタック、および関連リソースを作成および管理することで、変換された .NET アプリケーションをデプロイできます。

説明

このポリシーには、以下のアクセス許可が含まれています。

- CloudFormation – AWSTransform で始まる名前の CloudFormation スタックの作成、更新、削除、説明を許可します。スタックオペレーションは CreatedBy: AWSTransform でタグ付けされたリソースに制限され、同じ AWS アカウントに限定されます。
- Amazon EC2 – VPCs、サブネット、セキュリティグループ、イメージ、インスタンス、ルートテーブル、インターネットゲートウェイの説明を許可します。CloudFormation を介して呼び出された場合にのみ、EC2 インスタンスの実行、開始、停止、終了、変更を許可します。タグの作成は、CloudFormation オペレーション中にのみ、特定の許可されたタグキーに制限されます。
- AWS Identity and Access Management (IAM) – AWSTransform デプロイインスタンスの特定の IAM ロールを取得して渡すことを許可します。ロールポリシーと添付ファイルを検査するアクセス許可が含まれています。アクセスは同じ AWS アカウントに制限されています。
- Amazon EC2 Systems Manager (SSM) – AWS マネージドパラメータストアから Amazon Linux AMI パラメータを取得し、AWSTransform タグ付きインスタンスにコマンドを送信できるようにします。
- Amazon S3 – バケットの一覧表示や同じ AWS アカウント内のバケットの場所の取得など、AWSTransform デプロイバケット内のオブジェクトの管理を許可します。
- AWS Key Management Service (KMS) – AWSTransform にタグ付けされた KMS キーを使用した暗号化および復号オペレーションを許可します。S3 および EC2 サービスの使用に制限があります。

このポリシーは、リソースレベルのアクセス許可、タグベースの条件、を使用したサービスコントロールの制限 `aws:CalledVia`、アカウントレベルの制限、明示的な拒否ステートメントを通じて最小特権アクセスを実装し、CloudFormation オペレーション以外での不正なタグ変更を防ぎます。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSTransformApplicationDeploymentPolicy](#)」を参照してください。

AWS マネージドポリシー: AWSTransformApplicationECSDeploymentPolicy

このポリシーにより、AWS Transform は ECS クラスター、サービス、タスク、および関連リソースを作成および管理することで、変換されたアプリケーションを Amazon ECS にデプロイできます。

説明

このポリシーには、以下のアクセス許可が含まれています。

- CloudFormation – AWSTransform で始まる名前の CloudFormation スタックの作成、更新、削除、説明を許可します。スタックオペレーションは CreatedBy: AWSTransform でタグ付けされたリソースに制限され、同じ AWS アカウントに限定されます。
- Amazon ECS – ECS クラスター、サービス、タスク定義の作成、更新、削除を許可します。実行中のタスク、タスクの一覧表示、タスクステータスの記述を許可します。すべてのオペレーションは、AWSTransform で始まり、CreatedBy: AWSTransform でタグ付けされた名前のリソースに制限されます。
- AWS Identity and Access Management (IAM) – ECS タスク (AWSTransform-Deploy-ECS-Task-Role および AWSTransform-Deploy-ECS-Execution-Role) の特定の IAM ロールを取得して渡すことを許可します。ロールポリシーを検査し、必要に応じて ECS サービスにリンクされたロールを作成するアクセス許可が含まれています。
- Amazon CloudWatch Logs – /aws/ecs/AWSTransform で始まる名前のロググループの作成、削除、管理を許可します。デプロイされたアプリケーションをトラブルシューティングするためのログイベントの取得を許可します。
- Amazon ECR – アプリケーションコンテナイメージを保存するための awstransform で始まる名前のコンテナリポジトリの作成を許可します。
- AWS Key Management Service (KMS) – カスタマー管理の KMS キーを使用する場合に、ECR 暗号化用の許可の作成とデータキーの生成を許可します。

このポリシーは、リソースレベルのアクセス許可、タグベースの条件、アカウントレベルの制限を通じて最小特権アクセスを実装し、オペレーションが同じ AWS アカウント内の AWSTransform マネージドリソースに制限されるようにします。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSTransformApplicationECSDeploymentPolicy](#)」を参照してください。

AWS マネージドポリシー: AWSTransformCustomFullAccess

このポリシーは、AWS 変換カスタムへのフルアクセスを提供します。

説明

このポリシーには、以下のアクセス許可が含まれています。

- AWS Transform Custom – すべての AWS Transform カスタムリソースに対するすべてのアクションを許可します。これにより、サービスへの完全な管理アクセスが提供されます。
- AWS Identity and Access Management (IAM) – AWS Transform カスタム[サービスにリンクされたロール](#) () の作成を許可します `AWSServiceRoleForAWSTransformCustom`。このロールは、AWS 変換カスタムが CloudWatch メトリクスとログをアカウントに出力するために必要です。アクセス許可は、この特定のサービスにリンクされたロールの作成のみを許可するようにスコープされています。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSTransformCustomFullAccess](#)」を参照してください。

AWS マネージドポリシー: AWSTransformCustomExecuteTransformations

このポリシーは、AWS 変換カスタムで変換を実行するためのアクセスを提供します。

説明

このポリシーには、以下のアクセス許可が含まれています。

- AWS カスタム変換 – ストリーミング会話、変換の実行、キャンペーンの管理を許可します。キャンペーンの詳細の取得、キャンペーンリポジトリのステータスの更新、キャンペーンの更新を行うアクセス許可が含まれます。
- AWS Identity and Access Management (IAM) – AWS 変換カスタム[サービスにリンクされたロール](#) () の作成を許可します `AWSServiceRoleForAWSTransformCustom`。このロールは、AWS 変換カスタムが CloudWatch メトリクスとログをアカウントに出力するために必要です。アクセス許可は、この特定のサービスにリンクされたロールの作成のみを許可するようにスコープされています。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSTransformCustomExecuteTransformations](#)」を参照してください。

AWS マネージドポリシー: AWSTransformCustomManageTransformations

このポリシーは、AWS 変換カスタムで変換リソースを作成、更新、読み取り、削除したり、変換を実行したりするためのアクセスを提供します。

説明

このポリシーには、以下のアクセス許可が含まれています。

- AWS カスタム変換 – ストリーミング会話、変換の実行、変換パッケージの管理など、変換リソースを包括的に管理できます。変換パッケージ URLs の作成、取得、削除、およびパッケージのアップロードの完了を許可します。
- ナレッジ管理 – ナレッジ項目とその設定とステータスを一覧表示、取得、削除、更新できます。
- キャンペーン管理 – キャンペーンの詳細の取得、キャンペーンリポジトリのステータスの更新、キャンペーンの更新を許可します。
- リソースタグ付け – AWS Transform カスタムリソースのタグの一覧表示、追加、削除を許可します。
- AWS Identity and Access Management (IAM) – AWS 変換カスタム[サービスにリンクされたロール](#) () の作成を許可します `AWSServiceRoleForAWSTransformCustom`。このロールは、AWS 変換カスタムが CloudWatch メトリクスとログをアカウントに出力するために必要です。アクセス許可は、この特定のサービスにリンクされたロールの作成のみを許可するようにスコープされています。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSTransformCustomManageTransformations](#)」を参照してください。

AWS マネージドポリシー: AWSTransformInfrastructureExecutorAccessBatch

このポリシーは、AWS Batch を使用して AWS Transform Continuous Modernization の評価と変換を実行するために必要なアクセス許可を付与します。このポリシーをアカウントの IAM ロールにア

タッチします。AWS Transform Continuous Modernization CLI とエージェントはこのロールを引き受けて、ソースコードのアップロード、結果の取得、バッチジョブのステータスのモニタリング、ログの読み取り、変換スケジュールの管理を行います。

説明

このポリシーには、以下のアクセス許可が含まれています。

- AWS Lambda – 変換ワークフローを調整する atx プレフィックスが付いた Lambda 関数の設定の呼び出しと取得を許可します。
- Amazon S3 – atx-source-code-* バケットへのソースコードのアップロードと、atx-custom-output-* および atx-ct-output-* バケットからの変換結果のダウンロードを許可します。
- AWS Key Management Service (KMS) – AWS 変換暗号化キー () を使用したデータの暗号化と復号を許可します alias/atx-encryption-key。
- Amazon CloudWatch Logs – 変換のモニタリングとデバッグのためのバッチジョブと Lambda 実行ログイベントの読み取りを許可します。
- Amazon CloudWatch – オペレーションを可視化するための AWS Transform CLI ダッシュボードの取得と一覧表示を許可します。
- AWS バッチ – コンピューティング環境、ジョブキュー、ジョブ定義、ジョブステータスの記述と一覧表示を許可します。
- CloudFormation – AWS Transform インフラストラクチャスタックを記述および一覧表示して、デプロイステータスを確認できるようにします。
- Resource Groups Tagging API – タグ付けされたリソースを取得して、アカウント内の AWS 変換インフラストラクチャを検出できるようにします。
- AWS Secrets Manager – atx/プレフィックスに保存されている AWS 変換シークレットの取得と記述を許可します。
- Amazon EventBridge スケジューラ – atx-ct スケジュールグループ内でスケジュールされた変換ジョブを管理できます。
- Amazon EC2 – バッチコンピューティング環境ネットワークの VPC、サブネット、セキュリティグループ、ルートテーブル、NAT ゲートウェイ設定の説明を許可します。
- AWS Identity and Access Management (IAM) – ATX プレフィックス付きおよび Atx プレフィックス付きロール設定の読み取りを許可してインフラストラクチャのセットアップを検証し、を Amazon EventBridge スケジューラ AtxSchedulerInvocationRole に渡して、スケジュールされたジョブを呼び出せるようにします。

このポリシーは、リソースレベルのアクセス許可、タグベースの条件、aws:ResourceAccount条件、サービス範囲iam:PassedToServiceの条件を通じて最小特権アクセスを実装し、オペレーションが同じ AWS アカウント内の AWS Transform リソースに制限されるようにします。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWSTransformInfrastructureExecutorAccessBatch](#)」を参照してください。

AWS マネージドポリシー: AWSTransformInfrastructureExecutorAccessEC2

このポリシーは、Amazon EC2 を使用して AWS Transform Continuous Modernization の評価と変換を実行するために必要なアクセス許可を付与します。このポリシーをアカウントの IAM ロールにアタッチします。AWS Transform Continuous Modernization CLI とエージェントはこのロールを引き受けて、ソースコードのアップロード、結果の取得、ジョブステータスのモニタリング、変換スケジュールの管理を行います。

説明

このポリシーには、以下のアクセス許可が含まれています。

- CloudFormation – インフラストラクチャスタックの AWS 変換、スタックイベント、スタックリソース、ドリフト検出ステータスの説明、スタックのリスト、デプロイ前の CloudFormation テンプレートの検証を許可します。
- Amazon EC2 – インスタンス、AMIs、VPCs、サブネット、セキュリティグループ、キーペア、ルートテーブル、NAT ゲートウェイ、インターネットゲートウェイの説明を許可します。でタグ付けされた EC2 インスタンスの開始と停止を許可しますatx-remote-infra: true。
- Amazon EC2 Systems Manager (SSM) – コマンド実行ステータスのモニタリング、およびAWS-RunShellScriptドキュメントの実行を含む AWS 変換タグ付きインスタンスでのコマンドとセッションの実行を許可します。
- Amazon S3 – オブジェクトの取得、配置、削除、バケットの一覧表示など、AWS 変換バケットでのソースコードと出力アーティファクトの管理を許可します。
- AWS Key Management Service (KMS) – AWS 変換暗号化キー () を使用したデータの暗号化と復号を許可しますalias/atx-encryption-key。
- AWS Secrets Manager – atx/プレフィックスに保存されているリポジトリ認証情報などの変換シークレットの取得と記述を許可します。
- Amazon EventBridge スケジューラ – atx-ct スケジュールグループ内でスケジュールされた変換ジョブを管理できます。

- Resource Groups Tagging API – タグ付けされたリソースを取得して、アカウント内の AWS 変換 インフラストラクチャを検出できるようにします。
- AWS Identity and Access Management (IAM) – `atx-transform-role` ロールを Amazon EC2 に渡し、を Amazon EventBridge スケジューラ `AtxSchedulerInvocationRole` に渡し、EC2 インスタンスのロールとインスタンスプロファイル設定を読み取ることを許可します。

このポリシーは、リソースレベルのアクセス許可、タグベースの条件、`aws:ResourceAccount` 条件、サービス範囲 `iam:PassedToService` の条件を通じて最小特権アクセスを実装し、オペレーションが同じ AWS アカウント内の AWS Transform リソースに制限されるようにします。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の [AWS Transform Infrastructure Executor Access EC2](#)」を参照してください。

AWS マネージドポリシー: AWSTransformSecurityAgentExecutorAccess

このポリシーは、コードセキュリティの自動レビューと修復のために AWS セキュリティエージェントサービスを呼び出すために必要なアクセス許可を AWS Transform Continuous Modernization に付与します。AWS Transform Continuous Modernization CLI とエージェントがスキャンアーティファクトをアップロードして結果を取得するために引き受けるアカウントの IAM ロールにこのポリシーをアタッチします。

説明

このポリシーには、以下のアクセス許可が含まれています。

- AWS セキュリティエージェント – エージェントスペースの一覧表示、コードレビューの作成、コードレビュージョブの開始、検出結果の一覧表示と取得、コード修復の開始を許可します。これらのアクションは、アカウントのエージェントスペースに限定されます。
- Amazon S3 – `kct-security-agent-*` バケットからスキャン結果と検出結果をダウンロードし、セキュリティスキャン用のソースコードをそのバケット内の `security-scans/*` プレフィックスにアップロードできるようにします。
- AWS Identity and Access Management (IAM) – スキャンを実行できるように、AWS Security Agent サービスに `security-agent-*` ロールを渡すことを許可します。

このポリシーは、リソースレベルのアクセス許可、S3 バケットとキープレフィックス、`aws:ResourceAccount` および `s3:ResourceAccount` 条件、およびサービス範

囲iam:PassedToServiceの条件を通じて最小特権アクセスを実装し、オペレーションが同じ AWS アカウント内の Security Agent リソースに制限 AWS されるようにします。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[AWS Transform Security Agent Executor Access](#)」を参照してください。

AWS マネージドポリシー: DBModDiscoveryAndAssessment

このポリシーは、AWS 変換のための包括的なデータベースモダナイゼーションの検出および評価機能を提供します。

説明

このポリシーには、以下のアクセス許可が含まれています。

- Amazon EC2 – インスタンス、VPCs、サブネット、セキュリティグループ、アベイラビリティーゾーン、VPC エンドポイント、インターネットゲートウェイなどのインフラストラクチャコンポーネントの説明を許可します。
- Amazon RDS – データベースインスタンス、クラスター、サブネットグループの説明を許可します。移行の準備のために、同じ AWS アカウント内の DB サブネットグループの変更を許可します。
- Amazon RDS Data API – HTTP エンドポイントの有効化と無効化、およびデータベースモダナイゼーションプロジェクト用にタグ付けされたデータベースクラスターでの SQL ステートメントの実行を許可します。
- AWS DMS – エンドポイント、レプリケーションインスタンス、タスク、サブネットグループ、および順序付け可能なインスタンスの記述を許可します。データプロバイダー、インスタンスプロファイル、移行プロジェクトの一覧表示を許可します。テーブル統計、評価の実行、メタデータモデルオペレーションの説明を許可します。詳細な説明とメタデータオペレーションは、データベースモダナイゼーションプロジェクト用にタグ付けされたリソースに制限されます。
- – データベース認証情報を検出するためのシークレットの一覧表示を許可します。
- AWS Identity and Access Management (IAM) – 特定の AWS DMS サービスロールとそのアタッチされたポリシーの検査を許可します。読み取り AWS 管理 AWS DMS ポリシーへのアクセスが含まれます。
- AWS Key Management Service (KMS) – キーエイリアスの一覧表示とキーの説明を許可します。同じ AWS アカウントに制限された統合によるシークレットの復号を許可します。

このポリシーは、リソースレベルのアクセス許可、タグベースの条件、アカウントレベルの制限を通じて最小特権アクセスを実装し、オペレーションが同じ AWS アカウント内のデータベースモダナイゼーションプロジェクトリソースに制限されるようにします。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、AWS「マネージドポリシーリファレンスガイド」の[DBModDiscoveryAndAssessment](#)」を参照してください。

AWS マネージドポリシー: DBModProvisioningAndMigration

このポリシーは、データベースモダナイゼーションプロジェクトの AWS 変換のためのデータベースのプロビジョニングと移行機能を提供します。これには、移行インフラストラクチャの作成と管理、ターゲットデータベースのプロビジョニング、移行データの保存を行うためのアクセス許可が含まれています。

説明

このポリシーには、以下のアクセス許可が含まれています。

- AWS DMS – レプリケーションサブネットグループ、インスタンスプロファイル、データプロバイダー、移行プロジェクト、エンドポイント、レプリケーションインスタンス、レプリケーションタスクの作成と管理を許可します。メタデータモデルのインポート、変換、エクスポート、評価などのスキーマ変換オペレーションが含まれます。レプリケーションインスタンス (作成、削除、変更、再起動) およびレプリケーションタスク (削除、開始、停止、評価) のライフサイクル管理を許可します。すべての書き込みオペレーションは、データベースモダナイゼーションプロジェクト用にタグ付けされたリソースに制限されます。
- Amazon RDS – データベースサブネットグループ、データベースクラスター、データベースインスタンスの作成を許可します。データベースモダナイゼーションプロジェクトでは、すべてのリソースにタグを付ける必要があります。
- – データベースモダナイゼーションの命名プレフィックスを使用して、シークレットの作成、更新、タグ付けを許可します。シークレット値の取得と、タグ付けされたリソースのシークレットの説明を許可します。
- Amazon S3 – 移行データストレージ用の S3 バケットとオブジェクトの作成と管理を許可します。バケットのタグ付け、バージョニング、オブジェクトのライフサイクルオペレーションが含まれます。データベースモダナイゼーションの命名プレフィックスを持つバケットに制限されます。
- AWS Identity and Access Management (IAM) – 特定の AWS DMS サービスロールを AWS DMS およびスキーマ変換サービスに渡すことを許可します。データベースオペレーションに必要な Amazon RDS サービスにリンクされたロールの作成を許可します。

このポリシーは、リソースレベルのアクセス許可、タグベースの条件、アカウントレベルの制限を通じて最小特権アクセスを実装し、オペレーションが同じ AWS アカウント内のデータベースモダライゼーションプロジェクトリソースに制限されるようにします。

アクセス許可の詳細

ポリシーのアクセス許可の詳細については、「AWS マネージドポリシーリファレンスガイド」の[DBModProvisioningAndMigration](#)」を参照してください。

AWS 変換アクセス許可リファレンス

このセクションでは、AWS Transform で使用される APIs とその動作について説明します。

トピック

- [AWS 変換 APIs](#)

AWS 変換 APIs

- transform:BatchGetMessage
- transform:BatchGetUserDetails
- transform:CompleteArtifactUpload
- transform>CreateArtifactDownloadUrl
- transform>CreateArtifactUploadUrl
- transform>CreateAssetDownloadUrl
- transform>CreateConnector
- transform>CreateFeedback
- transform>CreateJob
- transform>CreateSession
- transform>CreateWorkspace
- transform>DeleteConnector
- transform>DeleteJob
- transform>DeleteSelfRoleMappings
- transform>DeleteUserRoleMappings
- transform>DeleteWorkspace

- transform:DetectIsAllowedForOperation
- transform:GetConnector
- transform:GetFeedbackQuestions
- transform:GetHitITask
- transform:GetJob
- transform:GetLoginRedirectUri
- transform:GetUserDetails
- transform:GetUserPreferences
- transform:GetWebAppUrl
- transform:GetWorkspace
- transform:ListArtifacts
- transform:ListConnectors
- transform:ListHitITasks
- transform:ListJobPlanSteps
- transform:ListJobs
- transform:ListMessages
- transform:ListPlanUpdates
- transform:ListUserRoleMappings
- transform:ListWorklogs
- transform:ListWorkspaces
- transform:PutUserPreferences
- transform:PutUserRoleMappings
- transform:RevokeSession
- transform:SearchUsers
- transform:SearchUsersTypeahead
- transform:SendMessage
- transform:StartJob
- transform:StopJob
- transform:SubmitCriticalHitITask

- transform:SubmitStandardHitlTask
- transform:TestReleaseTraitPreProd
- transform:TestReleaseTraitProd
- transform:UpdateHitlTask
- transform:UpdateJob
- transform:UpdateWorkspace
- transform:VerifySession

AWS 変換のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、「コンプライアンス[AWS のサービス プログラムによるスコープ](#)」の「コンプライアンス」を参照して、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、[AWS 「コンプライアンスプログラム」](#)を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「[Downloading Reports in AWS Artifact](#)」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。を使用する際のコンプライアンス責任の詳細については AWS のサービス、[AWS 「セキュリティドキュメント」](#)を参照してください。

AWS 変換の耐障害性

AWS グローバルインフラストラクチャは、AWS リージョン およびアベイラビリティゾーンを中心に構築されています。は、低レイテンシー、高スループット、高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティゾーン AWS リージョン を提供します。アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェールオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、フォールトトレランス、および拡張性が優れています。

AWS リージョン およびアベイラビリティゾーンの詳細については、[AWS 「グローバルインフラストラクチャ」](#)を参照してください。

AWS グローバルインフラストラクチャに加えて、AWS Transform には、データの耐障害性とバックアップのニーズをサポートするのに役立つ機能がいくつか用意されています。

AWS Transform およびインターフェイスエンドポイント (AWS PrivateLink)

VPC と の間にプライベート接続を確立するには、インターフェイス VPC エンドポイント AWS Transform を作成します。インターフェイスエンドポイントは、インターネットゲートウェイ [AWS PrivateLink](#)、NAT デバイス、VPN 接続、または Direct Connect 接続なしで AWS Transform コンソールにプライベートにアクセスできるテクノロジーである を利用しています。VPC と の間のトラフィック AWS Transform は、Amazon ネットワークを離れません。

各インターフェイスエンドポイントは、サブネット内の 1 つ、または複数の [Elastic Network Interface](#) によって表されます。

詳細については、「Amazon VPC ユーザーガイド」の「[インターフェイス VPC エンドポイント \(AWS PrivateLink\)](#)」を参照してください。

Note

AWS Transform カスタム PrivateLink ドキュメントについては、「」を参照してください [AWS Transform カスタムエンドポイントとインターフェイスエンドポイント \(AWS PrivateLink\)](#)。

AWS Transform VPC エンドポイントに関する考慮事項

のインターフェイス VPC エンドポイントを設定する前に AWS Transform、Amazon VPC ユーザーガイドの [インターフェイスエンドポイントのプロパティと制限](#)を確認してください。

前提条件

以下の手順を開始する前に、次の項目が揃っていることを確認してください。

- リソースを作成および設定するための適切なアクセス許可を持つ AWS アカウント。
- AWS アカウントに既に作成された VPC。
- AWS サービス、特に Amazon VPC と に精通していること AWS Transform。

のインターフェイス VPC エンドポイントの作成 AWS Transform

Amazon VPC コンソールまたは AWS Command Line Interface () を使用して、AWS Transform サービスの VPC エンドポイントを作成できます AWS CLI。詳細については、「Amazon VPC ユーザーガイド」の [インターフェイスエンドポイントの作成](#) を参照してください。

AWS Transform は、次の VPC エンドポイントサービスをサポートしています。

サービス名	Endpoint	メモ
コントロールプレーン	com.amazonaws. <i>region</i> .transform	
エージェント API	com.amazonaws. <i>region</i> .transform-agents	
ウェブアプリケーション	com.amazonaws. <i>region</i> .api.transform	AWS Transform ウェブアプリケーションに必要です。 このエンドポイントでは、 が VPC 内のプライベート IP アドレスに api.transform. <i>region</i> .on.aws 解決されるように、プライベート DNS が有効になっている必要があります (DNS 名を有効にするオプション)。

region を、com.amazonaws.us-east-1.transform など、プロファイルがインストールされている AWS リージョンに置き換えます AWS Transform。

Note

AWS Transform ウェブアプリケーションを使用する場合は、api.transform エンドポイントが必要です。完全なセットアップガイドについては、「」を参照してください [VPC から AWS Transform ウェブアプリケーションにアクセスする](#)。

詳細については、「Amazon VPC ユーザーガイド」の[でサポートされているリージョン AWS Transform](https://docs.aws.amazon.com/vpc/latest/userguide/vpce-interface.html#access-service-though-endpoint)「」および「インターフェイスエンドポイントを介したサービスへのアクセス」を参照してください。<https://docs.aws.amazon.com/vpc/latest/userguide/vpce-interface.html#access-service-though-endpoint>

オンプレミスコンピュータを使用して AWS Transform エンドポイントに接続する

このセクションでは、オンプレミスコンピュータを使用して AWS VPC の AWS PrivateLink エンドポイント AWS Transform を介してに接続するプロセスについて説明します。

1. [オンプレミスデバイスと VPC の間に VPN 接続を作成します。](#)
2. [のインターフェイス VPC エンドポイントを作成します AWS Transform。](#)
3. [インバウンド Amazon Route 53 エンドポイントをセットアップします。](#)これにより、オンプレミスデバイスからエンドポイントの DNS AWS Transform 名を使用できます。

VPC から AWS Transform ウェブアプリケーションにアクセスする

AWS PrivateLink を使用して VPC から AWS Transform API にプライベートにアクセスする場合、ウェブアプリケーションには追加のネットワーク設定が必要です。インターネット接続を必要とする CloudFront を介して静的コンテンツ (HTML、JavaScript、CSS) を提供します。ウェブアプリケーションからの API コールは VPC エンドポイントを通過し、完全にプライベートのままになります。

このガイドでは、VPC から制御されたインターネット出力を設定して、VPC を必要なドメインのみにロックしたままウェブアプリケーションがロードできるようにする方法について説明します。

仕組み

AWS Transform ウェブアプリケーションは 2 つのネットワークパスを使用します。

- API コール – ウェブアプリケーション (ジョブの開始、ワークスペースの表示など) を操作すると、ブラウザは API リクエストを `api.transform.region.on.aws` に送信します。 `api.transform.region.on.aws.com.amazonaws.region.api.transform` VPC エンドポイントとプライベート DNS を有効にすると、これらのリクエストは VPC 内のプライベート IP アドレスに解決され、AWS ネットワークを離れることはありません。
- 静的コンテンツ – ウェブアプリケーションの HTML、JavaScript、および CSS ファイルは、`tenant-id.transform.region.on.aws` を介して CloudFront を介して提供されます。 CloudFront

コンテンツ配信はパブリックインターネットでのみ利用できるため、これらのファイルをロードするにはインターネット接続が必要です。

- 認証 – AWS IAM Identity Center のサインインフローは `region.signin.aws` を使用します。これにはインターネット接続も必要です。

セキュリティを維持しながらウェブアプリケーションを有効にするには、AWS Network Firewall とドメインベースのフィルタリングを使用して、制御された出力パスを作成します。これにより、VPC は他のすべてのインターネットトラフィックをブロックしながら、ウェブアプリケーションに必要な特定のドメインにのみ到達できます。

アーキテクチャ

次の図は、ウェブアプリケーショントラフィックのネットワークパスを示しています。

```
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 は、TLS SNI 検査を実行するために、トラフィックの両方向 (対称ルーティング) を確認する必要があります。プライベートサブネット宛てのトラフィックをファイアウォール経由で送信するリターンルートは NAT Gateway サブネットで設定する必要があります。これを行わないと、ドメインベースのフィルタリングルールは機能しません。

前提条件

開始する前に、以下があることを確認してください。

- VPC リソース、Network Firewall、NAT Gateway を作成するアクセス許可を持つ AWS アカウント。
- インスタンスまたはワークロードが実行されるプライベートサブネットを持つ VPC。
- VPC にアタッチされたインターネットゲートウェイ (または作成するためのアクセス許可)。
- プライベート DNS が有効になってい `com.amazonaws.region.api.transform` の AWS Transform VPC エンドポイント。これは、ウェブアプリケーションブラウザクライアントで使われるエンドポイントです。エンドポイントの作成手順については、「」を参照してください [AWS Transform およびインターフェイスエンドポイント \(AWS PrivateLink\)](#)。

VPC エンドポイントポリシーの設定

`com.amazonaws.region.api.transform` VPC エンドポイントにカスタムエンドポイントポリシーがある場合は、AWS Transform オペレーションを許可するステートメントを追加する必要があります。これを行わないと、ウェブアプリケーションはエンドポイントを介して API コールを実行できなくなります。

VPC エンドポイントポリシーに次のステートメントを追加しま

す。 `AWS_TRANSFORM_PROFILE_ARN` を、設定ページの AWS Transform コンソールにある AWS Transform サービスプロファイル ARN に置き換えます。

```
{
  "Statement": [
    {
      "Sid": "AllowAWSTransform",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalArn": [
            "AWS_TRANSFORM_PROFILE_ARN"
          ]
        }
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Note

VPC エンドポイントがデフォルトのポリシー (フルアクセス) を使用している場合は、変更は不要で、このステップをスキップできます。

制御されたインターネット出力の設定

AWS Transform ウェブアプリケーションのドメインベースのフィルタリングを使用して Network Firewall を設定するには、次の手順を実行します。

ステップ 1: ファイアウォールサブネットを作成する

Network Firewall エンドポイント専用の小さな /28 サブネットを作成します。このサブネットは、プライベートサブネットと同じアベイラビリティゾーンにある必要があります。

```
aws ec2 create-subnet \  
  --vpc-id your-vpc-id \  
  --cidr-block firewall-subnet-cidr \  
  --availability-zone your-az \  
  --region region
```

ステップ 2: NAT ゲートウェイのパブリックサブネットを作成する

```
aws ec2 create-subnet \  
  --vpc-id your-vpc-id \  
  --cidr-block public-subnet-cidr \  
  --availability-zone your-az \  
  --region region
```

ステップ 3: パブリックサブネットルートテーブルを設定する

インターネットトラフィックをインターネットゲートウェイにルーティングするパブリックサブネットのルートテーブルを作成します。

```
# 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
```

ステップ 4: NAT ゲートウェイを作成する

```
# 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
```

ステップ 5: Network Firewall ルールグループを作成する

AWS Transform ウェブアプリケーションに必要なドメインへのトラフィックのみを許可するステートフルルールグループを作成します。

```
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
```

region を、プロファイルがインストールされている AWS リージョン (など) に置き換えます AWS Transform us-east-1。 *directory-id* を IAM Identity Center ディレクトリ ID に置き換えます (例: d-1234567890)。ディレクトリ ID は IAM Identity Center コンソールで確認できます。

次の表に、許可されたドメインを示します。

ドメイン	目的
.cloudfront.net	CloudFront CDN – ウェブアプリケーションの静的アセット (JavaScript、CSS、イメージ) を提供します
.transform. <i>region</i> .on.aws	ウェブアプリケーションテナント URL – ブラウザは CloudFront を介してこのドメインから最初のページをロードします
<i>region</i> .signin.aws	SSO サインインリダイレクトページ
.s3. <i>region</i> .amazonaws.com	S3 署名付き URLs – アーティファクトのアップロードとダウンロード
oidc. <i>region</i> .amazonaws.com	SSO 認証のための OIDC トークン交換
portal.sso. <i>region</i> .amazonaws.com	SSO ポータルのログインページ
assets.sso-portal. <i>region</i> .amazonaws.com	SSO ポータルの静的アセット (CSS、JavaScript)
<i>directory-id</i> .awsapps.com	組織の IAM Identity Center ポータル

Note

.cloudfront.net ワイルドカードは、AWS Transform ウェブアプリケーションのだけでなく、任意の CloudFront ディストリビューションへのトラフィックを許可します。CloudFront エッジ IPs はディストリビューション間で共有され、TLS SNI 検査では同じドメインの背後にある個々のディストリビューションを区別できないため、ドメインフィルターを絞り込むことはできません。

Note

をapi.transform.*region*.on.aws経由する API コール AWS PrivateLink。インターネット出力は必要ありません。ファイアウォールの影響を受けません。

ステップ 6: ファイアウォールポリシーを作成する

ポリシーでは、デフォルトのアクション `drop_established` として `STRICT_ORDER` ルール評価を使用する必要があります。これにより、許可リストと一致しないトラフィックはすべて削除されます。

```
# 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
```

ステップ 7: Network Firewall を作成する

```
# 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
```

ステップ 8: ファイアウォールエンドポイント ID を取得する

```
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"
```

ステップ 9: ファイアウォールサブネットルートテーブルを設定する

インターネットにバインドされたトラフィックをファイアウォールサブネットから NAT ゲートウェイにルーティングします。

```
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
```

ステップ 10: プライベートサブネットルートテーブルを更新する

プライベートサブネットからインターネットにバインドされたすべてのトラフィックをファイアウォール経由でルーティングします。

```
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
```

デフォルトのルートが既に存在する場合は、`replace-route` の代わりに `create-route` を使用します。

ステップ 11: 対称リターンルートを追加する (必須)

Important

このステップは重要です。Network Firewall は TLS SNI 検査を使用し、TCP 接続の両方向を確認する必要があります。プライベートサブネット宛てのリターントラフィックをファイアウォール経由で送信するルートを NAT Gateway サブネットのルートテーブルに追加します。

```
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
```

```
--route-table-id $PUB_RTB \  
--destination-cidr-block private-subnet-cidr \  
--vpc-endpoint-id $FW_ENDPOINT \  
--region region
```

private-subnet-cidr をプライベートサブネットの CIDR ブロック (例:) に置き換えます10.0.144.0/20。

対称ルーティングがない場合、ファイアウォールはトラフィックの一方向のみを表示します。TLS 検査エンジンは TLS ハンドシェイクからサーバー名表示 (SNI) を抽出できず、すべてのドメインベースのルールはサイレントに失敗します。

ステップ 12: IPv6 デフォルトルートを削除する (存在する場合)

プライベートサブネットルートテーブルの IPv6 デフォルトルート (:::/0) がインターネットゲートウェイを直接指している場合、IPv6-capableの送信先のファイアウォールをバイパスします。削除します。

```
aws ec2 delete-route \  
  --route-table-id private-subnet-route-table-id \  
  --destination-ipv6-cidr-block ::/0 \  
  --region region
```

検証

プライベートサブネットのインスタンスから、設定を確認します。

```
# 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)
```

トラブルシューティング

api.transform への API コールはファイアウォールによってブロックされます

ドメイン `api.transform.region.on.aws` は VPC エンドポイントを介してプライベート IP アドレスに解決され、インターネットファイアウォールに到達しないでください。

- `com.amazonaws.region.api.transform` エンドポイントが作成されていることを確認します。
- エンドポイントでプライベート DNS が有効になっていることを確認します。

```
aws ec2 describe-vpc-endpoints \  
  --filters "Name=service-name,Values=com.amazonaws.region.api.transform" \  
  --query 'VpcEndpoints[*].[State,PrivateDnsEnabled]' \  
  --output table
```

正常な出力: available | True

ウェブアプリケーションがロードされない (接続タイムアウト)

- プライベートサブネットルートテーブルにファイアウォールエンドポイントを指す `0.0.0.0/0` ルートがあることを確認します。
- ファイアウォールサブネットルートテーブルに NAT ゲートウェイを指す `0.0.0.0/0` ルートがあることを確認します。
- NAT ゲートウェイが available 状態であることを確認します。

許可されていないドメインはブロックされません

- インターネットゲートウェイを指す IPv6 `:::/0` ルートを確認します。これにより、ファイアウォールがバイパスされます。削除します (ステップ 12)。
- 対称ルーティングが設定されていることを確認します。NAT Gateway サブネットルートテーブルには、プライベートサブネット CIDR のファイアウォール経由のリターンルートが必要です (ステップ 11)。
- ファイアウォールポリシーが `drop_established` と `STRICT_ORDER` をデフォルトアクション `alert_established` として使用していることを確認します。

コストに関する考慮事項

[リソース]	概算コスト
Network Firewall	アベイラビリティーゾーンあたり最大 0.395 USD/時間 (最大 288 USD/月)
NAT Gateway	~0.045 USD/時間 (~33 USD/月) + データ処理料金
Elastic IP (パブリック IPv4)	~0.005 USD/時間 (~3.60 USD/月)
Network Firewall データ処理	0.065 USD/GB
NAT Gateway データ処理	0.045 USD/GB

1 つのアベイラビリティーゾーンデプロイの推定ベースコストは約 325 USD/月です。本番デプロイでは、プライベートサブネットが存在する各アベイラビリティーゾーンにファイアウォール、NAT ゲートウェイ、および関連するサブネットをデプロイします。

クリーンアップ

このガイドで作成されたすべてのリソースを削除するには、次のコマンドを逆の順序で実行します。プレースホルダーをデプロイIDs に置き換えます。これらの値は、AWS マネジメントコンソールで、またはセットアップステップの `describe` コマンドを使用して確認できます。

```
# 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
```

AWS 変換のモニタリング

モニタリングは、AWS Transform およびその他の AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。は、AWS Transform を監視し、問題が発生したときに報告し、必要に応じて自動アクションを実行するための以下のモニタリングツール AWS を提供します。

- Amazon CloudWatch は、AWS リソースと、で実行しているアプリケーションを AWS リアルタイムでモニタリングします。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、および指定したメトリクスが指定したしきい値に達したときに通知またはアクションを実行するアラームの設定を行うことができます。例えば、CloudWatch で Amazon EC2 インスタンスの CPU 使用率などのメトリクスを追跡し、必要に応じて新しいインスタンスを自動的に起動できます。詳細については、「[Amazon CloudWatch ユーザーガイド](#)」を参照してください。
- Amazon CloudWatch Logs では、Amazon EC2 インスタンス、CloudTrail、その他ソースから得たログファイルのモニタリング、保存、およびアクセスが可能です。CloudWatch Logs は、ログファイル内の情報をモニタリングし、特定のしきい値が満たされたときに通知します。高い耐久性を備えたストレージにログデータをアーカイブすることも可能です。詳細については、「[Amazon CloudWatch Logs ユーザーガイド](#)」を参照してください。
- Amazon EventBridge を使用して AWS サービスを自動化し、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。AWS サービスからのイベントは、ほぼリアルタイムで EventBridge に配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。詳細については、「[Amazon EventBridge ユーザーガイド](#)」を参照してください。
- AWS CloudTrail は、AWS アカountによって、またはアカウントに代わって行われた API コールおよび関連イベントをキャプチャし、指定した Amazon S3 バケットにログファイルを配信します。呼び出し元のユーザーとアカウント AWS、呼び出し元の送信元 IP アドレス、呼び出しの発生日時を特定できます。詳細については、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

を使用した AWS 変換 API コールのログ記録 AWS CloudTrail

AWS Transform は AWS CloudTrail、Transform のユーザー、ロール、または サービスによって実行されたアクションを記録する AWS サービスである AWS と統合されています。CloudTrail は、AWS Transform のすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、AWS 変換コンソールからの呼び出しと AWS、変換 API オペレーションへのコード

呼び出しが含まれます。証跡を作成する場合は、AWS 変換のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、AWS Transform に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

CloudTrail の詳細については、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

AWS CloudTrail での情報の変換

CloudTrail は、アカウントの作成 AWS アカウント 時に で有効になります。AWS Transform でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「[CloudTrail イベント履歴でのイベントの表示](#)」を参照してください。

Transform のイベントなど AWS アカウント、 のイベントの継続的な記録については AWS、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。詳細については、次を参照してください:

- [追跡を作成するための概要](#)
- [CloudTrail がサポートされているサービスと統合](#)
- 「[CloudTrail の Amazon SNS 通知の設定](#)」
- [複数のリージョンから CloudTrail ログファイルを受け取る](#) および [複数のアカウントから CloudTrail ログファイルを受け取る](#)

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。

- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「[CloudTrail userIdentity エlement](#)」を参照してください。

AWS 変換ログファイルエントリについて

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは、任意の出典からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

通知

AWS Transform は、ワークスペースのアクセス変更、ジョブステータスの更新、保留中の共同作業者のリクエストに関する通知を E メールで提供します。

E メール通知

E メール通知はデフォルトで有効になっており、ジョブに対する読み取り専用以上のアクセス許可を持つユーザーに送信されます。ユーザーは通知設定を変更できます。E メール通知には 3 つのカテゴリがあります。

- Workspace アクセスの更新

新しいワークスペースに追加したとき、およびワークスペース内でロールが変更されたときに通知が送信されます。

- 毎日のダイジェスト E メール

毎日太平洋標準時午前 9 時にダイジェスト E メールが届きます。この E メールには、1 つ以上の未解決の必須共同作業者リクエストを持つアクティブなジョブがある場合に、進行中のジョブがまとめられています。E メールには、ワークスペース名、ジョブ名、ジョブステータス、共同作業者リクエストの詳細、および必要なアクション (10 行以内) を示すテーブルが含まれています。

Note

日次ダイジェスト E メールは、必要な共同作業者リクエストがあるジョブに対してのみ送信されます。オプションの共同作業者リクエストは、ダイジェスト E メールをトリガーしません。

- ジョブステータスの更新

アクセス可能なジョブが完了または失敗すると、E メールが送信されます。

- コラボレーターリクエストの更新

タスクに入力、レビュー、または承認が必要なときはいつでも E メールが届きます。

E メール通知の管理

ウェブアプリケーションの E メール通知設定を変更するには、アプリケーションの右上隅にある設定コグをクリックし、通知設定を選択します。

通知設定ページには、次のコントロールが表示されます。

通知タイプ	説明	デフォルト設定
Workspace アクセスの更新	ワークスペースに追加したとき、またはワークスペース内でロールが変更されたときの Eメールの受信	有効
日次コラボレーターリクエストダイジェスト	アクセス可能なワークスペースのジョブに対するオープンな共同作業者リクエストのダイジェストを毎日受け取る	有効
ジョブ完了の更新	ジョブが完了または失敗したときに E メールを受信する	有効
コラボレーターリクエストの更新	入力または承認が必要な場合に E メールを受信する	Disabled

一般的な設定を使用して、すべての通知を有効または無効にしたり、個別の通知タイプを個別に設定したりできます。

AWS 変換のクォータ

AWS アカウントには、AWS サービスごとに、以前は制限と呼ばれていたデフォルトのクォータがあります。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。

AWS 変換のクォータを表示するには、[Service Quotas コンソール](#)を開きます。ナビゲーションペインで、AWS サービスを選択し、AWS 変換を選択します。

クォータの引き上げをリクエストするには、Service Quotas ユーザーガイドの「[クォータ引き上げリクエスト](#)」を参照してください。Service Quotas でクォータがまだ利用できない場合は、[制限引き上げフォーム](#)を使用します。

AWS アカウントには、AWS 変換に関連する次のクォータがあります。

ワークロード	ディメンション	クォータ	調整可能	説明
プラットフォーム	AWS Transform 有効なアカウント AWS リージョンごとのワークスペース	サポートされているワークスペースごとに 100 個 AWS リージョン	はい	対応 AWS Transform アカウントあたりのワークスペースの最大数 AWS リージョン
VMWare 移行	各ウェーブのサーバー数	各ウェーブ: 150 サーバー	いいえ	ウェーブ内のサーバーの最大数
メインフレーム	各 IAM Identity Center アカウントの同時ジョブ	サポートされているリージョンごとに 10 個のジョブ	はい	IAM Identity Center アカウントの同時メインフレームジョブの最大数
	各ジョブのコード行	サポートされているリージョンごとに	いいえ	各ジョブのコードの最大行数。

ワークロード	ディメンション	クォータ	調整可能	説明
		3,000,000 行のコード		
	毎月のコード行の分析	サポートされているリージョンごとに 100,000,000 行のコード	はい	1 暦月におけるコード分析ジョブ目標コードの最大行数
	ビジネスドキュメント生成の毎月のコード行	サポートされているリージョンごとに 50,000,000 行のコード	はい	1 暦月におけるビジネスドキュメント生成ジョブ目標コードの最大行数
	技術文書生成の毎月のコード行	サポートされているリージョンごとに 50,000,000 行のコード	はい	1 暦月におけるテクニカルドキュメント生成ジョブ目標コードの最大行数
	コード行の変換	サポートされているリージョンごとに 10,000,000 行のコード	はい	1 暦月に変換されたコードの最大行数
	毎月のコード行の分解	サポートされているリージョンごとに 50,000,000 行のコード	はい	1 暦月における分解ジョブ目標コードの最大行数

ワークロード	ディメンション	クォータ	調整可能	説明
	毎月のコード行のリフォージ	サポートされているリージョンごとに 50,000,000 行のコード	はい	1 暦月におけるリフォージジョブ目標コードの最大行数
	抽出されたビジネスロジックの月次コード行	サポートされているリージョンごとに 50,000,000 行のコード	はい	暦月内の抽出されたビジネスロジックジョブ目標のコードの最大行数
.NET	.NET の月ごとのコード行 (ウェブと IDE)	サポートされているリージョンごとに 1,000,000 行のコード	はい	IAM Identity Center アカウントでサポートされている各で暦月に変換できる .NET AWS リージョン アプリケーションコードの最大行数
	各 IAM Identity Center アカウントの Microsoft Visual Studio Extension との .NET ジョブの同時実行	サポートされているリージョンごとに 10 個のジョブ	はい	Microsoft Visual Studio 拡張機能を使用する場合には IAM Identity Center アカウント AWS リージョンでサポートされている各の同時 .NET 変換ジョブの最大数。

ワークロード	ディメンション	クォータ	調整可能	説明
	各 IAM Identity Center アカウントの同時 .NET ウェブジョブ	サポートされているリージョンごとに 5 つのジョブ	はい	Transform ウェブアプリケーションを使用する場合 AWS、IAM Identity Center アカウント AWS リージョンでサポートされている各の同時 .NET 変換ジョブの最大数。
カスタム変換	カスタム変換定義	1,000	はい	カスタム変換定義の最大数
	カスタム変換定義サイズ	10 MB	いいえ	ドキュメント参照を含む、変換定義に含まれるすべてのファイルの最大合計サイズ。
	カスタム変換の毎月の CLI 会話リクエスト	1,000,000	はい	1 暦月に CLI で実行できるカスタム変換会話リクエストの最大数
評価	各評価のサーバー数制限	各評価: 30,000 サーバー	いいえ	1 つの評価ジョブ内のサーバーの最大数

でサポートされているリージョン AWS Transform

Note

このページに記載されていないオプトインリージョンから情報を取得 AWS Transform する必要があるリクエストを行うと、はそのリージョンを呼び出す AWS Transform ことができます。を呼び出す AWS Transform ことができるリージョンへのアクセスを管理するには、「」を参照してください[AWS 変換におけるセキュリティ](#)。

このトピックでは、を使用できる AWS リージョンについて説明します AWS Transform。の詳細については AWS リージョン、「AWS アカウント管理 リファレンスガイド」の[AWS リージョン「アカウントで使用できるを指定する」](#)を参照してください。

データは、使用するリージョンとは異なるリージョンで処理される場合があります AWS Transform。でのクロスリージョン処理の詳細については AWS Transform、「」を参照してください[クロスリージョン処理](#)。処理中にデータが保存される場所については、[データ保護](#)に関する説明を参照してください。

Supported AWS リージョン (デフォルトで有効)

AWS Transform ワークスペースは、以下で作成できます AWS リージョン。これらのリージョンはデフォルトで有効になっています。使用前に有効にする必要はありません。詳細については、[デフォルトで有効になっているリージョン](#)の説明を参照してください。

- 米国東部 (バージニア北部)
- 欧州 (フランクフルト)
- アジアパシフィック (ムンバイ)
- アジアパシフィック (シドニー)
- アジアパシフィック (東京)
- 欧州 (ロンドン)
- アジアパシフィック (ソウル)
- カナダ (中部)
- 南米 (サンパウロ) - メインフレームモダナイゼーションエージェントのみ

ジョブを作成するワークスペースによって、ジョブ AWS リージョン のが決まります。別のリージョンでジョブを作成するには、目的のリージョンにある別のワークスペースを使用する必要があります。

VMware プロジェクトでは、ワークスペースのリージョンが検出に使用されます。ただし、移行ターゲットとして別のリージョンを指定できます。このターゲットリージョンは、移行の完了時にワークロードがホストされる場所です。可能なターゲットリージョンのリストなど、VMware 移行に関する AWS リージョン 考慮事項の詳細については、「」を参照してください [the section called “サポートされているターゲットリージョン”](#)。

AWS のサポート AWS Transform

AWS Transform では、変換ジョブ中に問題が発生した場合に、チャットを通じて AWS サポート ケースを作成および管理できます。ケースはジョブコンテキストで自動的に強化され、適切なサポート チームにルーティングされます。アクティブな Countdown Premium のお客様は、指定されたエンジニアが問題解決を管理するために連絡します。詳細については[AWS](#)、[「Countdown Premium」](#)を参照してください。そうしないと、サポートチームが AWS Transform サポートキューでケースを調査します。

前提条件

サポートケース管理を使用する前に、以下が必要です。

- アクティブな AWS サポートプラン - Business、Enterprise On-Ramp、Enterprise
- 既存の AWS Transform アプリケーション

サポートケース管理の有効化

1. AWS Transform コンソールを開きます。
2. ナビゲーションペインで [設定] を選択します。
3. サポート で、サポートケース管理を有効にする を選択します。
4. [Save changes] (変更の保存) をクリックします。

AWS Transform は、サービスにリンクされたロール (SLR) を使用して、ユーザーに代わってサポートケースを作成および管理します。追加の IAM アクセス許可は必要ありません。

サポートケースの作成

サポートケースを作成するには、チャットインターフェイスを使用します。

1. サポートケースを作成するよう に依頼します。例えば、次のようになります。
 - 「失敗した変換ジョブのサポートケースを作成する」
 - 「ジョブエラーに関するヘルプが必要、ケースを作成する」
2. チャットでは、必要な情報の入力を求められます。

- 問題の簡単な説明
- 詳細な説明

3. プロンプトが表示されたら、ケースの作成を確認します。

ケースには、以下を含む変換ジョブのコンテキストが自動的に入力されます。

- ジョブ ID とステータス
- ジョブメタデータ
- ジョブ実行ログ (作業ログ)
- ジョブプラン

ケースは、ジョブタイプと問題カテゴリに基づいて自動的に割り当てられます。

サポートケースの表示

サポートケースを表示するには、チャットにプロンプトを表示します。

- 「サポートケースを一覧表示する」
- 「オープンサポートケースを表示する」
- 「自分のケースのステータスは何ですか？」

チャットには以下が表示されます。

- ケース ID
- 件名
- オープン、保留中、解決済み、クローズなどのステータス
- 最終更新日
- 関連付けられたジョブ ID

特定のケースの詳細を表示するには:

- 「ケース [case-id] の詳細を表示する」
- 「ケース [case-id] の最新の更新は何ですか？」

ケースへの通信の追加

既存のケースにメッセージを追加するには:

1. チャットに尋ねる: 「ケース [case-id] にメッセージを追加する」
2. プロンプトが表示されたら、メッセージを入力します。
3. チャットは、メッセージが追加されたことを確認します。

AWS サポートがケースに応答すると、通知が送信されます。

ケースの解決

サポートケースを解決するには:

1. チャットに質問する: 「ケースを解決する [case-id]」
2. プロンプトが表示されたら、解決を確認します。

Note

AWS サポートによって対処されたケースのみを解決できます。

サポートケース管理の無効化

1. AWS Transform コンソールを開きます。
2. ナビゲーションペインで [設定] を選択します。
3. サポートで、サポートケース管理を無効にするを選択します。
4. [Save changes] (変更の保存) をクリックします。

Note

サポートケース管理を無効にすると、チャットベースのケース管理インターフェイスは削除されますが、既存のケースは閉じられません。サポート AWS センターからケースにアクセスできます。

使用状況テレメトリ

AWS Transform IDE プラグイン、AWS Transform Agent Skill、および AWS Transform Kiro Power は、変換の実行中にデフォルトで使用状況テレメトリを収集します。テレメトリは、IDE 名 (VS Code や Kiro など)、AI エージェント名 (Claude Code や OpenAI Codex など)、実行モード (ローカルまたはリモート) など、さまざまなデータポイントで構成されます。このデータは、互換性テストとレイテンシーと信頼性を優先するために AWS Transform によって使用されます。

テレメトリを無効にするには、チャットセッション中にテレメトリを出力しないようにエージェントに指示します。このオプトアウトは現在のチャットセッションにのみ適用され、新しいセッションごとに繰り返す必要があります。AWS Transform CLI を直接実行している場合は、`ATX_DISABLE_TELEMETRY=true`環境変数を設定します。

運用データ

当社では、サービスを提供し、サービスのパフォーマンスを評価するために、自然言語のフィードバックに基づいて、使用する機能、エラーログ、メトリクスなど、AWS Transform の使用に関する運用上のシグナルを収集することがあります (変換計画を進めることを承認したか、提案された計画を反復する手順を提供したかなど)。

変更ログ

次の表に、AWS Transform の注目すべきリリースと更新を示します。

日付	タイトル	説明	Links
2026-07-16	AWS Transform for Migrations がダウンロード可能なワークスペース概要レポートを追加	AWS Transform for Migrations では、ワークスペースの概要レポートをダウンロード可能な PDF として生成できるようになりました。このレポートは、ジョブのステータス、ワークフローステップの進行状況、ウェーブプランニング、ネットワーク移行、ランディングゾーン設定、リホストの進行状況、生成されたアーティファクトなど、ワークスペース内のすべての移行ジョブのデータを統合します。	ワークスペースの概要レポートのダウンロード
2026-07-03	AWS 変換検出ツールで Oracle Database のサポートを追加	AWS Transform 検出ツールは、VMware、Hyper-V、ベアメタルサーバー全体のデータベースから直接詳細な Oracle メタデータを収集するようにな	Oracle Database の検出

日付	タイトル	説明	Links
		りました。Oracle インスタンスは、直接 SQL 接続を通じて検出できます。このツールは、Container Database/Pluggable Database (CDB/PDB) の列挙、コンポーネントインベントリ、データファイルのサイジングを収集します。また、実際のアプリケーションクラスター (RAC)、データガード、マルチテナントアーキテクチャのトポロジフラグもキャプチャします。データベース認証情報を設定していない場合、ツールは OS レベルの検出に戻ります。すべての Oracle バージョンがサポートされています。	

日付	タイトル	説明	Links
2026-06-22	南米 (サンパウロ) リージョンでのメインフレームモダライゼーションエージェントのサポート	AWS メインフレームモダライゼーションエージェントの変換が南米 (サンパウロ) リージョン (sa-east-1) で利用可能になりました。サポートされているリージョンとクロスリージョン推論テーブルに sa-east-1 を追加しました。	サポートされているリージョン、クロスリージョン推論
2026-06-22	AWS VMware 移行の変換がローカリゼーションをサポート	AWS Transform for VMware 移行がローカリゼーションをサポートするようになりました。VMware 移行ワークフローを操作するときにウェブアプリケーションの表示言語を変更するには、設定アイコンを選択し、任意の言語を選択します。	ローカリゼーション設定

日付	タイトル	説明	Links
2026-06-18	AWS Transform for Migrations が移行ターゲットとしてすべての AWS 商用リージョンをサポート	AWS VMware 移行の変換は、中東 (バーレーン) と中東 (アラブ首長国連邦) を除くすべての AWS 商用リージョンを移行ターゲットとしてサポートします。移行ターゲットリージョンは、ランディングゾーン、ネットワークインフラストラクチャ、サーバーのリホストなど、移行されたリソースがデプロイされる AWS リージョンです。	サポートされている移行ターゲットリージョン
2026-06-18	AWS Transform for Migrations では、既存の ENIs を起動テンプレートにアタッチできます。	AWS Transform を使用すると、移行用の起動テンプレートに既存の Elastic Network Interface (ENIs) をアタッチできます。AWS Transform は、ターゲットアカウントで見つかった既存の ENIs をすべて表示し、インスタスの起動時に起動設定で使えるようにタグ付けできます。	移行用の ENI リソースのタグ付け

日付	タイトル	説明	Links
2026-06-18	AWS Transform for Migrations では、レプリケーションと起動の設定を行うことができます。	AWS 変換を使用すると、移行のレプリケーション設定と起動設定を設定できます。ターゲットアカウントごと、または特定のソースサーバー間で設定を設定できます。	レプリケーションと起動の設定

日付	タイトル	説明	Links
2026-06-17	生成 AI ワークロードの Model-to-model 移行評価	AWS Transform は、model-to-model 移行のカスタム変換を提供するようになりました。変換は生成 AI ワークロードを評価します。サードパーティープロバイダーから Amazon Bedrock に移行するための移行計画を作成します。AI を活用したエージェントはコードベースをスキャンして、使用中のすべての AI SDK とモデルを識別します。エージェントは、インタラクティブな質問を通じて移行要件を収集します。次に、透過的なコスト比較と本番環境に対応したコード変更により、モデルを Amazon Bedrock の同等のものにマッピングします。LiteLLM または Ollama を介した OpenAI、Google Gemini、Anthropic SDK の直接使用、オープンソースモデルから	最新情報の投稿、カスタム変換

日付	タイトル	説明	Links
		の移行をサポートします。LangChain、Llama Index、CrewAI、LangGraph、マルチプロバイダールーティングレイヤーなど、直接 SDK 統合とフレームワークラップパターンを処理します。標準の AWS Transform 料金を超える追加料金なしで利用できます。	

日付	タイトル	説明	Links
2026-06-16	AWS メインフレームの変換で追跡可能な再考ワークフローを実現	AWS メインフレームの変換は、評価からコード生成まで、接続され追跡可能な再考エクスペリエンスを提供するようになりました。z/OS COBOL および PL/I ワークロードを実行している企業は、ポートフォリオを評価して、個別のビジネス機能を特定し、ビジネスルールを抽出し、開発に対応した要件を生成し、単一の接続ワークフローで追跡可能なクラウドネイティブコードを生成できます。すべての要件は、完全な監査可能性のためにソースコードにさかのぼります。	最新情報の投稿、ドキュメント

日付	タイトル	説明	Links
2026-06-16	AWS Transform がターゲットストレージとして Amazon FSx for NetApp ONTAP をサポートするようになりました (パブリックプレビュー)	コンピューティングリホストの一環として、Amazon EBS に加えて、ブロックストレージワークロードを Amazon FSx for NetApp ONTAP に移行できるようになりました。AWS Transform は、コンピューティングとネットワークを処理するのと同じ移行ウェーブ内で、ブロックストレージデータを FSx for ONTAP ボリュームに直接レプリケートします。	最新情報の投稿、ドキュメント

日付	タイトル	説明	Links
2026-06-05	Amazon RDS for SQL Server の TCO 評価のサポート	オンプレミスの SQL Server データベースを Amazon RDS for SQL Server に移行するときコストを見積もることができるようになりました。Transform の AI を活用した TCO 評価 AWS を使用して環境を分析し、最適なデータベースインスタンスのレコメンデーションを取得します。この評価では、Bring Your Own Media (BYOM) および License Included (LI) オプションと Database Savings Plans がサポートされており、最大 20% の節約になります。	最新情報の投稿、ドキュメント

日付	タイトル	説明	Links
2026-06-04	AWS 変換検出ツールがネットワーク移行入力ソースとしてサポートされるようになりました	AWS 変換検出ツールを modelizeIT とともにネットワーク移行のソースとして使用できるようになりました。これにより、VMware ワークロードと非 VMware ワークロードの両方を実行している環境のハイブリッドネットワーク移行が可能になります。	ユーザーガイド
2026-05-27	AWS 変換カスタムドキュメントでクライアント側のスキルの例とベストプラクティスを追加	クライアント側のスキルドキュメントには、Dockerfile コンプライアンス、データベース移行の安全性、Terraform ポリシーの適用、API 非推奨ヘルパーの実用的な例と、スキルの構築と共有に関するベストプラクティスが含まれています。	ドキュメント

日付	タイトル	説明	Links
2026-05-22	AWS Transform で新しいエージェント移行評価機能が利用可能に	AWS Transform は、What-If シナリオ、カスタマイズ可能な前提条件、柔軟なファイル形式のサポート、複数の新しい総所有コスト (TCO) 評価機能など、高度な移行評価機能を提供するようになりました。	最新情報の投稿、ドキュメント
2026-05-20	AWS 変換はネットワーク移行のレビュー中に既存の VPCs を検出します	AWS Transform VPCs を検出するようになりました。マッピング VPCs と一緒に既存の VPCs を表示し、CIDR の競合を特定し、デプロイ前に解決できます。	ドキュメント
2026-05-14	AWS Kiro、Claude、Cursor、および Codex で利用可能な変換エージェント	AWS 変換エージェントに Kiro 電源、エージェントプラグイン、および AWS 変換 MCP サーバーを介してアクセスできるようになりました。これにより、開発者は希望する開発環境から直接変換機能を使用できます。	最新情報の投稿

日付	タイトル	説明	Links
2026-05-14	AWS Transform は、カスタマイズされた変換エージェントを構築するための エージェントビルダー ツールキットを導入します。	パートナーとお客様は、エージェントビルダーツールキット Kiro パワーを使用して特殊な変換エージェントを構築、共有、登録し、カスタムツールとワークフローを AWS Transform と統合できるようにになりました。	最新情報の投稿
2026-05-14	AWS Transform が AWS Toolkit for Visual Studio にエージェント AI アシスタントを追加	.NET 開発者は、Visual Studio で会話型の step-by-stepのガイド付きエクスペリエンスを通じてアプリケーションをモダナイズできるようになりました。フルコンテキストは IDE とウェブコンソール全体で保持されます。	最新情報の投稿
2026-05-14	AWS Transform が 顧客所有のアーティファクトストアをサポートするようになりました	独自の Amazon S3 バケットを設定し、オプションで独自の KMS AWS キーを使用してアーティファクトを暗号化することで、変換アーティファクトの保存場所を完全に制御できます。	最新情報の投稿 、 ユーザーガイド

日付	タイトル	説明	Links
2026-05-14	AWS Transform が移行戦略 (7R) の推奨事項を追加	AWS Transform を使用すると、検出されたインベントリを分析し、サーバーとアプリケーションごとに移行戦略の推奨事項を取得できるようになりました。推奨事項は、業界標準の 7 つの移行戦略 (7R) に従います。各レコメンドーションには、推奨される AWS ターゲットサービス、信頼スコア、およびその背後にある推論が含まれます。インタラクティブな HTML ダッシュボード、PDF、または Microsoft PowerPoint 出力を生成できます。	7Rsレコメンドーション

日付	タイトル	説明	Links
2026-05-11	AWS Transform が移行中にコンテナ化機能を追加	AWS Transform では、移行中のアプリケーションのコンテナへのリプラットフォーム、Dockerfile 生成の自動化、CVE スキャンによるコンテナイメージビルド、Amazon ECS または Amazon EKS へのデプロイがサポートされるようになりました。	最新情報の投稿 、 ユーザーガイド
2026-05-01	AWS Transform が Power BI と Tableau から Quick への BI 移行エージェントを提供するようになりました	Wavicle Data Solutions のパートナー構築エージェントは、Power BI ダッシュボードと Tableau ダッシュボードを AWS 変換ワークフロー内のクイックアセットに変換します。	最新情報の投稿
2026-04-21	AWS 変換カスタムが 6 つの追加 AWS リージョンで利用可能に	AWS 変換カスタムは、アジアパシフィック、カナダ、欧州 (ロンドン) に拡張され、合計 8 つの AWS リージョンに達します。	最新情報の投稿

日付	タイトル	説明	Links
2026-04-15	AWS Transform が移行ワークフロー内にランディングゾーンの作成を追加	移行ワークフロー内でランディングゾーンの作成を直接追加しました。既存の基盤のチェックを AWS 変換し、組織単位 (OU) 構造とターゲットアカウントを推奨し、完全に自動化されたデプロイまたは Infrastructure as Code (IaC) 出力 (CloudFormation、AWS CDK、または Landing Zone Accelerator (LZA) 形式) を提供します。	最新情報の投稿、ドキュメント

日付	タイトル	説明	Links
2026-04-15	AWS Transform が移行計画図とレポートを追加	移行計画中にインタラクティブな図と分析レポートを生成できるようになりました。図タイプには、ネットワークポートマップ、アプリケーション依存関係グラフ、ウェーブガンツグラフ、一般グラフなどがあります。レポートタイプには、リスク評価、7Rs レコメンデーション、カスタム分析レポートが含まれます。出力は、インタラクティブな HTML、PDF、または Microsoft PowerPoint (.pptx) ファイルとして使用できます。	ダイアグラムとレポート
2026-04-14	AWS 変換が Kiro と VS Code で利用可能になりました	AWS 変換カスタム変換に Kiro と VS Code からアクセスできるようになりました。ジョブの状態は IDE、CLI、およびウェブコンソールの表面間で共有されます。	最新情報の投稿

日付	タイトル	説明	Links
2026-04-10	AWS Transform がネットワーク移行のセキュリティグループマッピングで DHCP を追加	ネットワーク移行中のセキュリティグループマッピングによる DHCP のサポートが追加されました。DHCP 対応の移行ワークロードがある場合、移行後もセキュリティルールが有効であることを確認できるようになりました。これは、VPC 内セキュリティグループ参照、Transit Gateway 経由のクロス VPC およびクロスアカウントアウトバウンドルールの両方に適用されます。	ドキュメント
2026-03-31	AWS Transform custom が自動コードベース分析の一般提供を発表	包括的なコードベース分析変換は、あらゆる言語でアーキテクチャドキュメント、技術的負債レポート、モダナイゼーションの推奨事項を生成し、一般公開を実現します。	最新情報の投稿

日付	タイトル	説明	Links
2026-03-31	AWS 変換カスタムは、コードを大規模にモダナイズするための新しいマネージド変換を導入します	Node.js アップグレード (一般提供)、Java 最適化、Log4j 移行、Angular-to-React、Angular アップグレード、Vue アップグレード (早期アクセス) を含む 7 つの新しいマネージド変換が追加されました。	最新情報の投稿
2026-03-20	AWS Transform APIs を追加	ネットワーク移行 APIs が追加され、パートナーとプログラムによるネットワーク移行機能へのアクセスが可能になりました。	API リファレンス
2026-01-14	AWS カスタム adds AWS PrivateLink サポートを変換し、欧州 (フランクフルト) に拡張	AWS Transform custom は、AWS PrivateLink を介したプライベート VPC アクセスをサポートするようになり、米国東部 (バージニア北部) に加えて欧州 (フランクフルト) でも利用可能になりました。	最新情報の投稿

日付	タイトル	説明	Links
2025-12-23	AWS 変換により、ハイブリッドデータセンター移行のネットワーク変換が可能に	AWS Transform for VMware は、手動でマッピングすることなく、ハイブリッドデータセンターネットワーク設定 (VLANs、IP 範囲) を AWS VPCs、サブネット、セキュリティグループに自動的に変換します。	最新情報の投稿
2025-12-01	AWS のローンチ AWS 組織全体のモダナイゼーションを加速するための変換カスタム	AWS 変換カスタムは一般公開されており、組織は単一の CLI コマンドを使用して反復可能なコード変換を大規模に自動化できます。	最新情報の投稿
2025-12-01	AWS Transform がエンタープライズ VMware 移行用の新しいエージェント AI 機能を追加	AWS Transform の VMware エージェントは、オンプレミス環境の検出、依存関係のマッピング、移行ウェブプランの生成、HyperV、Nutanix、KVM、ベアメタルターゲット間の移行が可能になりました。	最新情報の投稿

日付	タイトル	説明	Links
2025-12-01	AWS Transform がフルスタックの Windows モダナイゼーション用の AI エージェントを起動	新しいフルスタック Windows エージェントは、.NET アプリケーションと SQL Server データベースの Aurora PostgreSQL への変換を自動化し、Amazon ECS または EC2 Linux のコンテナにデプロイします。	最新情報の投稿
2025-12-01	AWS Transform は .NET 変換機能を拡張し、開発者エクスペリエンスを向上させます。	AWS Transform は、.NET 10 サポート、ASP.NET Web Forms to Blazor 移植、AWS Toolkit for Visual Studio による強化されたインタラクティブ IDE エクスペリエンスを追加します。	最新情報の投稿
2025-12-01	AWS メインフレームの変換が新しいテスト自動化機能を提供	AWS メインフレームの変換では、テストプランの自動生成、テストデータ収集スクリプト、テストケースの自動化が導入され、メインフレームのモダナイゼーションテストが高速化されます。	最新情報の投稿

日付	タイトル	説明	Links
2025-12-01	AWS メインフレームの変換がアプリケーションの再設計をサポートするようになりました	新しいデータとアクティビティの分析とビジネスロジックの抽出機能により、レガシーメインフレームアプリケーションをクラウドネイティブアーキテクチャに分解できます。	最新情報の投稿
2025-11-13	AWS Transform が Landing Zone Accelerator のネットワーク設定を自動化	AWS Transform for VMware は LZA 互換のネットワーク YAML ファイルを自動生成し、VMware 環境からのマルチアカウントクラウドインフラストラクチャのセットアップを合理化します。	最新情報の投稿
2025-07-16	AWS メインフレームの変換で、強化されたコードリファクタリング機能とビジネスロジック機能を導入	AWS メインフレームの変換により、リフォージ (コード再構築) の一般提供が追加され、アプリケーションレベルのビジネスロジック抽出が拡張され、コード品質が向上します。	最新情報の投稿

日付	タイトル	説明	Links
2025-05-15	AWS Transform for VMware が一般公開されました	AWS Transform for VMware は、VMware のモダナイゼーションライフサイクル全体を検出からデプロイまで自動化するエージェント AI サービスであり、一般公開されています。	最新情報の投稿
2025-05-15	AWS メインフレームの変換が一般利用可能になりました	AWS IBM z/OS アプリケーションを大規模にモダナイズする最初のエージェント AI サービスであるメインフレームのトランスフォームは、新しい分析、分解、ジョブ管理機能で一般公開されています。	最新情報の投稿

[このデータをダウンロード](#)します。

AWS 変換ユーザーガイドのドキュメント履歴

次の表に、AWS 変換ユーザーガイドのドキュメント履歴を示します。

変更	説明	日付
継続的なモダナイゼーション ポートフォリオの分析と修復	自動コード分析と修復のための AWS 継続的なモダナイゼーション変換機能 (atx ct サブコマンド) に関する新しい章を追加しました。 「継続的なモダナイゼーション」 を参照してください。	2026 年 6 月 3 日
ブラウフィールドネット ワーク: 既存の VPCs 可視性	ネットワーク移行レビュー中に既存の VPC 検出を追加しました。AWS Transform は、ターゲットアカウントに既にデプロイされている VPCs を自動的に検出し、マッピングされた VPCs。 VPCs 」を参照してください。	2026 年 5 月 18 日
AWS Transform custom のス キル形式、クライアント側の スキル、およびサブエージェ ントログを追加	変換定義でスキル形式を使用するようになりました (SKILL.mdYAML frontmatter とオプションの references/ フォルダを使用)。追加の機能を使用してエージェントを拡張するためのクライアント側のスキルと、デバッグを改善するためのサブエージェントログを追加しました。CLI バージョン 2.0 以降が必要です。 「変換の定義」	2026 年 5 月 11 日

と「[クライアント側のスキル](#)」を参照してください。

[ソースコードのコンテナ化を追加](#)

AWS 変換では、生成 AI を使用してアプリケーションをコンテナ化できるようになりました。Git リポジトリまたは zip アップロードからソースコードを提供し、AWS Transform がそれを分析して Docker アーティファクトを生成し、コンテナイメージを Amazon ECR に発行し、Infrastructure as Code を生成して、Amazon Elastic Container Service または Amazon Elastic Kubernetes Service にデプロイします。
[「ソースコードのコンテナ化」](#)を参照してください。

2026 年 5 月 7 日

[ネットワーク最適化オペレーションとガイド付きレコメンデーションを追加](#)

ステップ 5: VPC およびサブネットオペレーション (削除、除外、マージ、分割、サイズ変更、名前変更、IP アドレスの変更) と、命名標準化、スコープレビュー、容量の適切なサイズ設定、セキュリティレビュー、VPC 統合に関するガイド付きレコメンデーションを使用して、ネットワークを確認して最適化するを追加しました。
[「ネットワークの確認と最適化」](#)を参照してください。

2026 年 4 月 27 日

[AWS 6 つの新しいリージョン で利用可能なカスタムの変換](#)

[AWS Transform custom](#) で ca-central-1、eu-west-2、ap-northeast-1、ap-northeast-2、ap-southeast-2、ap-south-1 のサポートが追加されました。

2026 年 4 月 21 日

[コネクタのアクセス許可に関するドキュメントを更新](#)

S3、MGN、Migration Hub のアクセス許可、ワークロードの移行、ランディングゾーンインフラストラクチャのプロビジョニングなど、ターゲット AWS アカウント コネクタの詳細なアクセス許可の内訳を追加しました。[「ターゲットの接続 AWS アカウント」](#)を参照してください。

2026 年 4 月 19 日

[VMware 移行ジョブタイプを更新](#)

検出と移行の計画、ランディングゾーン、ランディングゾーン、ネットワーク、サーバー移行ジョブタイプを追加しました。Network migration only to Network migration and Migration planning to Build migration plan という名前に変更されました。[VMware 移行ジョブタイプ](#)」を参照してください。

2026 年 4 月 19 日

[再構築された Connect AWS アカウントターゲット](#)

コネクタのセットアップを移行タイプの選択、MAP 契約、コネクタ設定の 3 つのステップに再構成しました。これには、サポートされているターゲットリージョン、アカウントロール、委任管理者ガイダンス、IAM ロールが含まれます。[「ターゲットの接続 AWS アカウント」](#)を参照してください。

2026 年 4 月 19 日

[server-migration-onlyターゲットリージョンを追加](#)

ネットワーク移行なしでサーバー移行ジョブに使用できる商用リージョンが 14 個追加され、2026 年Q3 四半期まで許可リストの要件が文書化されました。[「サポートされているターゲットリージョン」](#)を参照してください。

2026 年 4 月 19 日

[サーバーの移行に関する章を追加](#)

アカウント構造、IAM ロール、委任管理者設定、レプリケーションエージェントのデプロイ、テスト、カットオーバーなど、リホストワークフロー全体をカバーする新しい[移行サーバーの](#)章を追加しました。

2026 年 4 月 19 日

[ビルドランディングゾーンの章を追加](#)

[ランディングゾーンプロビジョニングのコネクタアクセス許可の詳細を含む新しいビルドランディングゾーンの](#)章を追加しました。

2026 年 4 月 19 日

セキュリティグループのマッピングと参照	セキュリティグループマッピング戦略 (MAP、MAP_DHCP、SKIP) と、 ネットワークを移行するためのドキュメントを参照するセキュリティグループを追加しました。	2026 年 4 月 13 日
サポートされていないネットワークファイルタイプの自動変換	移行ネットワーク でサポートされていないネットワーク設定ファイルタイプの自動変換のサポートが追加されました。	2026 年 4 月 6 日
データベース移行用の KMS キーポリシーを追加	データベース移行の暗号化セクション を追加しました。	2026 年 3 月 26 日
文書化された新しい AWS 管理ポリシー	DBModProvisioningAndMigration セクションを追加しました。	2026 年 3 月 24 日
文書化された新しい AWS 管理ポリシー	DBModDiscoveryAndAssessment セクションを追加しました。	2026 年 3 月 24 日
ネットワーク図の生成	ネットワークの移行にオプションのネットワーク図の生成 (Mermaid 形式と PNG 形式) を追加しました。	2026 年 3 月 18 日
ファイアウォールと SDN 設定ファイルのサポート	移行ネットワークでのネットワーク移行のための Palo Alto Networks、Fortinet FortiGate、および Cisco ACI 設定ファイルのサポートが追加されました。 https://docs.aws.amazon.com/transform/latest/userguide/transform-vmware-migrate-network.html	2026 年 2 月 11 日

ネットワークリソースのカスタムタグ付け	移行ネットワークでのネットワーク移行に対するジョブレベルと VPC レベルのカスタムタグ付けのサポートを追加しました。	2026 年 1 月 28 日
サポートドキュメントを追加	セクション AWS のサポートを追加しました。	2025 年 12 月 23 日
文書化された新しい AWS 管理ポリシー	AWSTransformCustomFullAccess 、 AWSTransformCustomExecuteTransformations 、および AWSTransformCustomManageTransformations セクションを追加しました。	2025 年 12 月 7 日
VMware の移行	VMware 移行 セクションを更新しました。	2025 年 12 月 1 日
AWS 管理ポリシーの更新	AWSServiceRoleForAWSTransform ドキュメントを更新して、追加されたアクセス許可を反映しました。	2025 年 12 月 1 日
サポートされるリージョン	サポートされているリージョン のトピックを更新しました。	2025 年 12 月 1 日
メインフレームのモダナイゼーション	Mainframe modernization セクションを更新しました。	2025 年 12 月 1 日
開始方法	「開始方法 https://docs.aws.amazon.com/transform/userguide/getting-started.html 」トピックを更新しました。	2025 年 12 月 1 日

フルスタックの Windows モダナイゼーション	更新された .NET セクションと新しい SQL Server モダナイゼーションセクション を含むフルスタック Windows モダナイゼーショントピックを追加しました。 https://docs.aws.amazon.com/transform/userguide/sql-server-modernization.html	2025 年 12 月 1 日
検出ツール	検出ツール セクションを追加しました。	2025 年 12 月 1 日
AWS 管理ポリシーの更新	AWSTransformApplicationECSDeploymentPolicy を更新し、拡張された IAM ロール検査アクセス許可、ECS サービスにリンクされたロールの作成、ECR 暗号化サポートのための KMS アクセス許可を追加しました。	2025 年 11 月 22 日
AWS 管理ポリシーの更新	AWSTransformApplicationDeploymentPolicy を更新し、EC2 ネットワーク、IAM ロール検査、S3 バケット管理、KMS 暗号化のアクセス許可を追加しました。	2025 年 11 月 22 日
検出コレクター	AWS Transform discovery tool > ドキュメントを追加しました。	2025 年 10 月 31 日

VMware 移行アーキテクチャ図を追加	ネットワーク移行に VMware 移行アーキテクチャ図を追加しました。 https://docs.aws.amazon.com/transform/latest/userguide/transform-vmware-migrate-network.html	2025 年 10 月 23 日
VMware 移行結果の更新	移行 ネットワークの移行 結果情報を更新しました。	2025 年 10 月 12 日
VMware 移行結果の更新	「ネットワークを移行する」 の移行 結果情報を更新しました。	2025 年 10 月 12 日
S3 コネクタのドキュメント	.NET ウェブアプリケーションの S3 コネクタドキュメント を追加しました。	2025 年 10 月 8 日
新しいユーザーコネクタのトピック	ユーザーコネクタ のトピックを追加しました。	2025 年 9 月 21 日
AWS 管理ポリシーの更新	AWSServiceRoleForAWSTransform ドキュメントを更新しました。	2025 年 9 月 17 日
ストレージ評価のサポート	NetApp Data Infrastructure Insights (DII) ファイルによる ストレージ評価 のサポートが追加されました。	2025 年 9 月 2 日
ターゲットリージョンの更新	サポートされている 3 つのターゲットリージョン を追加しました。	2025 年 8 月 31 日

マネージドポリシーの更新	AWSServiceRoleForAWSTransform で使用される IAM アイデンティティセンターの読み取り専用権限を追加しました。	2025 年 8 月 29 日
文書化された新しい AWS 管理ポリシー	AWSTransformApplicationDeploymentPolicy セクションを追加しました。	2025 年 8 月 28 日
インターフェイスエンドポイント	インターフェイスエンドポイント のトピックを追加しました。	2025 年 7 月 9 日
初回リリース	AWS Transform ユーザーガイドの初回リリース。	2025 年 5 月 15 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。