



ユーザーガイド

AWS Clean Rooms



AWS Clean Rooms: 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

とは AWS Clean Rooms	1
初めての AWS Clean Rooms ユーザーですか?	1
の AWS Clean Rooms 仕組み	2
関連サービス	2
AWS サービス	2
サードパーティのサービス	4
アクセス AWS Clean Rooms	4
の料金 AWS Clean Rooms	4
の請求 AWS Clean Rooms	5
分析ルール	5
分析ルールの種類	6
集計分析ルール	9
リスト分析ルール	28
カスタム分析ルール	36
ID マッピングテーブル分析ルール	43
AWS Clean Rooms 差分プライバシー	53
差分プライバシー	54
での差分プライバシーの AWS Clean Rooms 仕組み	54
差分プライバシーポリシー	55
SQL 機能	57
SQL クエリのヒントと例	163
制限事項	164
AWS クリーンルーム ML	166
AWS Clean Rooms ML の用語	166
AWS Clean Rooms ML が AWS モデルと連携する方法	168
AWS Clean Rooms ML がカスタムモデルと連携する方法	169
AWS Clean Rooms ML のモデル	171
Clean Rooms ML のカスタムモデル	178
暗号コンピューティング	186
考慮事項	188
サポートされているファイルとデータの種類の種類	190
列名	195
列タイプ	196
パラメータ	198

オプションのフラグ	203
クエリと C3R	206
ガイドライン	207
セットアップ AWS Clean Rooms	233
にサインアップする AWS	233
のサービスロールを設定する AWS Clean Rooms	233
管理者ユーザーの作成	234
コラボレーションメンバー用の IAM ロールの作成	235
Amazon S3 からデータを読み取るサービスロールを作成する	236
Amazon Athena からデータを読み取るサービスロールを作成する	240
Snowflake からデータを読み取るサービスロールを作成する	243
S3 バケットからコードを読み取るサービスロールを作成する (PySpark 分析テンプレート ロール)	248
PySpark ジョブの結果を書き込むサービスロールを作成する	251
結果を受け取るサービスロールを作成する	254
AWS Clean Rooms ML のサービスロールを設定する	258
類似モデリングのサービスロールを設定する	258
カスタムモデリングのサービスロールを設定する	272
コラボレーションとメンバーシップ	288
コラボレーションの作成	289
クエリのコラボレーションの作成	289
クエリとジョブのコラボレーションの作成	300
ML モデリング用のコラボレーションの作成	311
メンバーシップの作成とコラボレーションへの参加	320
コラボレーションの編集	327
コラボレーションの名前と説明の編集	327
コラボレーション分析エンジンを更新する	328
ログストレージをオフにする	328
コラボレーションログの設定を編集する	329
コラボレーションのタグ編集	330
メンバーシップのタグ編集	331
新しいメンバーを追加する	331
既存のメンバー機能を編集する	332
コラボレーションの自動承認設定を編集する	332
関連付けられているテーブルのタグ編集	332
分析テンプレートのタグ編集	332

差分プライバシーポリシータグの編集	333
変更リクエスト	334
コラボレーションに新しいメンバーを追加する	334
既存のメンバー機能を更新する	335
コラボレーションの自動承認設定を編集する	336
コラボレーションの削除	337
コラボレーションの表示	338
メンバーの招待	338
メンバーのモニタリング	339
メンバーの追加	339
メンバーの削除	340
コラボレーションからの退出	341
データテーブル	343
データ形式	344
PySpark ジョブでサポートされているデータ形式	344
SQL クエリでサポートされているデータ形式	344
サポートされているデータ型	345
のファイル圧縮タイプ AWS Clean Rooms	346
のサーバー側の暗号化 AWS Clean Rooms	346
Apache Iceberg テーブル	347
Iceberg テーブルでサポートされているデータ型	348
データテーブルの準備	349
Amazon S3 でのデータテーブルの準備	349
Amazon Athena でのデータテーブルの準備	352
Snowflake でのデータテーブルの準備	354
暗号化データテーブルの準備	357
ステップ 1: 前提条件を満たす	357
ステップ 2: C3R 暗号化クライアントをダウンロードする	358
ステップ 3: (オプション) C3R 暗号化クライアントで使用可能なコマンドを表示する	359
ステップ 4: 表形式ファイルの暗号化スキーマを生成する	359
ステップ 5: 共有シークレットキーを作成する	366
ステップ 6: 共有シークレットキーを環境変数に保存する	367
ステップ 7: データを暗号化する	368
ステップ 8: データ暗号化を確認する	370
(オプション) スキーマの作成 (上級ユーザー)	371
データテーブルの復号化	380

設定済みテーブル	382
設定済みテーブルの作成	383
Amazon S3 データソース	383
Amazon Athena データソース	387
Snowflake データソース	390
設定済みテーブルへの分析ルールの追加	394
集計分析ルールのテーブルへの追加 (ガイドフロー)	395
リスト分析ルールのテーブルへの追加 (ガイドフロー)	398
カスタム分析ルールのテーブルへの追加 (ガイドフロー)	400
テーブルへの分析ルールの追加 (JSON エディタ)	404
次の手順	405
設定済みテーブルのコラボレーションへの関連付け	406
データアクセス予算	406
設定済みテーブルを関連付ける	407
Configuring a data access budget	413
データアクセス予算の表示	413
既存の関連テーブルへのデータアクセス予算の追加	414
データアクセス予算の編集	415
データアクセス予算の削除	417
設定済みテーブルへのコラボレーション分析ルールの追加	418
差分プライバシーポリシーの設定 (オプション)	419
差分プライバシー使用状況ログの表示	420
差分プライバシーポリシーの編集	421
差分プライバシーポリシーの削除	422
計算された差分プライバシーパラメータの表示	422
テーブルと分析ルールの表示	423
設定済みテーブルの編集	424
設定済みテーブルのタグの編集	424
設定済みテーブル分析ルールの編集	425
設定済みテーブル分析ルールの削除	426
設定済みテーブルの許可されていない列	426
設定済みテーブルの関連付けの編集	430
設定済みテーブルの関連付けの解除	430
AWS Entity Resolution in AWS Clean Rooms	432
ID 名前空間	433
新しい ID 名前空間の作成と関連付け	433

既存の ID 名前空間の関連付け	436
ID 名前空間の関連付けの編集	438
ID 名前空間の関連付け解除	439
ID マッピングテーブル	440
新しい ID マッピングテーブルの作成とデータ入力	441
既存の ID マッピングテーブルへのデータ入力	449
ID マッピングテーブルの編集	450
ID マッピングテーブルの削除	450
分析テンプレート	452
SQL 分析テンプレート	452
SQL 分析テンプレートの作成	453
SQL 分析テンプレートの確認	456
PySpark 分析テンプレート	457
セキュリティ	458
制限事項	458
ベストプラクティス	459
ユーザースクリプトの作成	460
PySpark 分析テンプレートでのパラメータの操作	464
仮想環境の作成 (オプション)	468
ユーザースクリプトと仮想環境を S3 に保存する	469
PySpark 分析テンプレートの作成	471
PySpark 分析テンプレートの確認	475
PySpark 分析テンプレートのトラブルシューティング	478
コードのトラブルシューティング	478
分析テンプレートジョブが開始されない	479
分析テンプレートジョブは開始されますが、処理中に失敗します	481
仮想環境の設定が失敗する	482
分析	484
SQL クエリの実行	484
前提条件	485
SQL クエリの Spark プロパティ設定	486
設定済みテーブルのクエリを実行する	487
ID マッピングテーブルのクエリを実行する	505
分析テンプレートを使用した設定済みテーブルのクエリ実行	507
分析ビルダーを使用してクエリを実行する	522
差分プライバシーの影響を表示する	528

最近のクエリの表示	529
クエリの詳細の表示	530
PySpark ジョブの実行	530
分析テンプレートを使用したジョブの実行	531
最近のジョブの表示	533
ジョブの詳細の表示	534
分析結果	536
クエリ結果の受信	537
ジョブ結果の受信	538
クエリ結果設定のデフォルト値の編集	539
ジョブ結果設定のデフォルト値の編集	541
他の のクエリ出力の使用 AWS のサービス	542
カスタムモデリング	543
プライバシー強化合成データセットの生成	544
合成データ生成に関する考慮事項	545
コラボレーションの作成と参加	548
機械学習用のコラボレーションの作成	548
コラボレーションへの参加	552
トレーニングデータの提供	554
モデルアルゴリズムの設定	559
設定されたモデルアルゴリズムの関連付け	562
ML 入力チャネルの作成	567
トレーニング済みモデルの作成	585
増分トレーニングの使用	588
分散トレーニングの使用	592
モデルアーティファクトのエクスポート	596
トレーニング済みモデルでの推論の実行	598
トレーニングデータプロバイダーの ML モデリング	601
トレーニングデータのインポート	602
類似モデルの作成	603
類似モデルの設定	604
設定された類似モデルの関連付け	605
設定された類似モデルの更新	606
シードデータプロバイダーの ML モデリング	608
類似セグメントの作成	608
類似セグメントのエクスポート	623

トラブルシューティング	624
クエリが参照する 1 つ以上のテーブルに、関連付けられたサービスロールでアクセスできない。テーブル/ロールの所有者が、サービスロールにテーブルへのアクセス許可を付与する必要がある。	624
基になるデータセットの 1 つに、サポートされていないファイル形式が使用されている。	624
Cryptographic Computing for Clean Rooms の使用時に、期待どおりのクエリ結果が得られない。	625
AWS Clean Rooms Spark SQL: パーティションデータがありません	625
セキュリティ	626
データ保護	627
保管中の暗号化	628
転送中の暗号化	629
基になるデータの暗号化	629
キーポリシー	629
サービスにリンクされたロールの使用	635
のサービスにリンクされたロールのアクセス許可 AWS Clean Rooms	636
のサービスにリンクされたロールの作成 AWS Clean Rooms	636
のサービスにリンクされたロールの編集 AWS Clean Rooms	637
のサービスにリンクされたロールの削除 AWS Clean Rooms	637
AWS Clean Rooms 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	638
データ保持	638
ベストプラクティス	639
を使用したベストプラクティス AWS Clean Rooms	639
で分析ルールを使用するためのベストプラクティス AWS Clean Rooms	639
Identity and Access Management	641
オーディエンス	642
アイデンティティを使用した認証	642
ポリシーを使用したアクセスの管理	644
が IAM と AWS Clean Rooms 連携する方法	646
アイデンティティベースのポリシーの例	652
AWS マネージドポリシー	655
トラブルシューティング	663
サービス間の混乱した代理の防止	665
AWS Clean Rooms ML の IAM 動作	667
クリーンルーム ML カスタムモデルの IAM 動作	671
コンプライアンス検証	673

耐障害性	673
インフラストラクチャセキュリティ	674
ネットワークセキュリティ	674
AWS PrivateLink	675
考慮事項	675
インターフェイスエンドポイントの作成	675
モニタリング	677
分析ログイン AWS Clean Rooms	678
クエリログとジョブログの受信	679
クエリログとジョブログに推奨されるアクション	680
での CloudWatch による詳細なモニタリング AWS Clean Rooms	680
メトリクスの種類	681
メトリクスのディメンション	682
メトリクスにアクセスできるユーザー	683
料金	683
CloudTrail ログ	683
AWS Clean Rooms CloudTrail の情報	683
AWS Clean Rooms ログファイルエントリについて	684
Example AWS Clean Rooms CloudTrail イベント	685
EDAs AWS Clean Rooms への統合	688
AWS Clean Rooms イベント	689
ルーティング AWS Clean Rooms イベント	694
イベントの詳細リファレンス	694
コスト配分タグ	704
コスト配分のタグ付け可能なリソース	704
AWS CloudFormation リソース	706
AWS Clean Rooms および CloudFormation テンプレート	706
の詳細 CloudFormation	706
クォータ	708
AWS Clean Rooms クォータ	708
AWS Clean Rooms リソースパラメータの制限	724
AWS Clean Rooms ML クォータ	724
Clean Rooms ML API スロットリングクォータ	762
ドキュメント履歴	768
用語集	783
集計分析ルール	783

分析ルール	783
分析テンプレート	783
C3R 暗号化クライアント	783
クリアテキスト列	784
コラボレーション	784
コラボレーションクリエイター	784
設定済みテーブル	785
カスタム分析ルール	785
復号	785
差分プライバシー	785
暗号化	786
フィンガープリント列	786
ID マッピングワークフローメソッド	786
ID マッピングテーブル	786
ID マッピングテーブル分析ルール	787
ID マッピングワークフロー	787
ID 名前空間	787
ID 名前空間の関連付け	787
ジョブ	787
リスト分析ルール	787
類似モデル	788
類似セグメント	788
メンバー	788
クエリを行えるメンバー	788
クエリとジョブを実行できるメンバー	788
結果を受け取れるメンバー	789
クエリの計算コストを負担するメンバー	789
クエリとジョブのコンピューティングコストを支払うメンバー	789
メンバーシップ	790
シール列	790
シードデータ	790
Spark 分析エンジン	790
クエリ	790
.....	dcxci

とは AWS Clean Rooms

AWS Clean Rooms は、お客様とパートナーが集合データセットを分析して共同作業を行い、基盤となるデータを相互に公開することなく、新しいインサイトを得ることができます。AWS Clean Rooms は安全なコラボレーションワークスペースであり、独自のクリーンルームを数分で作成し、集合データセットをわずか数ステップで分析できます。コラボレーションするパートナーを選択し、データセットを選択し、それらのパートナーのプライバシー強化コントロールを設定します。

を使用すると AWS Clean Rooms、既に使用している何千もの企業とコラボレーションできます AWS。コラボレーションでは、データを から移動 AWS したり、別のクラウドサービスプロバイダーにロードしたりする必要はありません。クエリまたはジョブを実行すると、はそのデータの元の場所からデータを AWS Clean Rooms 読み取り、組み込みの分析ルールを適用して、そのデータの制御を維持します。

AWS Clean Rooms には、設定できる組み込みのデータアクセスコントロールと監査サポートコントロールが用意されています。これらの制御には以下が含まれます。

- SQL クエリを制限し、出力制約を提供する [分析ルール](#)。
- [Cryptographic Computing for Clean Rooms](#) は、クエリが処理されてもデータを暗号化し、厳格なデータ処理ポリシーに準拠します。
- でクエリとジョブを確認し、監査のサポート AWS Clean Rooms に役立つ [分析ログ](#)。
- ユーザー識別の試みから保護するための [差分プライバシー](#)。AWS Clean Rooms 差分プライバシーは、数ステップで適用できる数学に基づく手法と直感的なコントロールにより、ユーザーのプライバシーを保護するフルマネージド機能です。
- [AWS Clean Rooms ML](#) を使用すると、2 つの当事者がデータ内の類似ユーザーを識別でき、相互にデータを共有する必要はありません。1 番目の関係者はトレーニングデータから類似モデルを作成し、設定します。次に、トレーニングデータに似た類似セグメントを作成するために、シードデータがコラボレーションに持ち込まれます。

次の動画では、の詳細について説明します AWS Clean Rooms。

[AWS Clean Rooms](#)

初めての AWS Clean Rooms ユーザーですか？

を初めて使用する場合は AWS Clean Rooms、まず以下のセクションを読むことをお勧めします。

- [の AWS Clean Rooms 仕組み](#)
- [アクセス AWS Clean Rooms](#)
- [セットアップ AWS Clean Rooms](#)
- [AWS Clean Rooms 用語集](#)

の AWS Clean Rooms 仕組み

で AWS Clean Rooms、コラボレーションを作成し、招待 AWS アカウント する を追加するか、招待されたコラボレーションに参加するメンバーシップを作成します。次に、ユースケースに必要なデータリソースである、イベントデータ用に設定されたテーブル、ML モデリング用に設定されたモデル、またはエンティティ解決用の ID 名前空間をリンクします。分析テンプレートを作成または承認して、コラボレーションで許可する正確なクエリとジョブについて事前に同意することができます。最後に、設定されたテーブルで SQL クエリまたは PySpark ジョブを実行するか、ID マッピングテーブルでエンティティ解決を実行するか、ML モデリングを使用して類似オーディエンスセグメントを生成することで、ジョイントデータを分析します。

次の図は、の AWS Clean Rooms 仕組みを示しています。

関連サービス

AWS サービス

以下は AWS のサービス、に関連しています AWS Clean Rooms。

- Amazon Athena

コラボレーションメンバーは、Amazon Athena の AWS Glue Data Catalog ビュー AWS Clean Rooms として取り込むデータを保存できます。詳細については、以下の各トピックを参照してください。

詳細については、以下の各トピックを参照してください。

[でのクエリ用のデータテーブルの準備 AWS Clean Rooms](#)

[設定済みテーブルの作成 – Amazon Athena データソース](#)

[Amazon Athena ユーザーガイドの「Amazon Athena とは Amazon Athena」](#)

- CloudFormation

で次のリソースを作成します CloudFormation: コラボレーション、設定済みテーブル、設定済みテーブルの関連付け、メンバーシップ

詳細については、「[を使用した AWS Clean Rooms リソースの作成 AWS CloudFormation](#)」を参照してください。

- AWS CloudTrail

CloudTrail ログ AWS Clean Rooms で を使用して、AWS のサービス アクティビティの分析を強化します。

詳細については、「[を使用した AWS Clean Rooms API コールのログ記録 AWS CloudTrail](#)」を参照してください。

- AWS Entity Resolution

AWS Clean Rooms で AWS Entity Resolution を使用してエンティティ解決を実行します。

詳細については、「[AWS Entity Resolution in AWS Clean Rooms](#)」を参照してください。

- AWS Glue

コラボレーションメンバーは、Amazon S3 のデータから AWS Glue テーブルを作成し、 で使用できます AWS Clean Rooms。

詳細については、以下の各トピックを参照してください。

[でのクエリ用のデータテーブルの準備 AWS Clean Rooms](#)

「AWS Glue デベロッパーガイド」の「[AWS Glueの概要](#)」

- Amazon Simple Storage Service (Amazon S3)

コラボレーションメンバーは、Amazon S3 AWS Clean Rooms に取り込むデータを保存できます。

詳細については、以下の各トピックを参照してください。

[でのクエリ用のデータテーブルの準備 AWS Clean Rooms](#)

[設定済みテーブルの作成 – Amazon S3 データソース](#)

Amazon Simple Storage Service デベロッパーガイドの [Amazon S3 とは？](#)

- AWS Secrets Manager

コラボレーションメンバーは、Snowflake に保存されているデータにアクセスして読み取るシークレットを作成できます。

詳細については、以下の各トピックを参照してください。

[Snowflake からデータを読み取るサービスロールを作成する](#)

[でのクエリ用のデータテーブルの準備 AWS Clean Rooms](#)

「AWS Secrets Managerユーザーガイド」の「[AWS Secrets Manager とは](#)」

サードパーティのサービス

以下のサードパーティーサービスが関連しています AWS Clean Rooms。

- Snowflake

コラボレーションメンバーは、取り込むデータを Snowflake ウェアハウス AWS Clean Rooms に保存できます。

詳細については、以下の各トピックを参照してください。

[でのクエリ用のデータテーブルの準備 AWS Clean Rooms](#)

[設定済みテーブルの作成 – Snowflake データソース](#)

アクセス AWS Clean Rooms

には、次のオプション AWS Clean Rooms を使用してアクセスできます。

- <https://console.aws.amazon.com/cleanrooms/> の AWS Clean Rooms コンソールから直接。
- AWS Clean Rooms API を使用してプログラムで。詳細については、「[AWS Clean Rooms API リファレンス](#)」を参照してください。

の料金 AWS Clean Rooms

料金に関する情報については、[\[AWS Clean Rooms の料金\]](#)を参照してください。

Note

Snowflake に保存されているデータを関連付けたコラボレーションメンバーの場合、それらの場所に保存されているデータを使用するクエリを実行するたびに、データ出力とコンピューティングの両方について、それぞれのデータウェアハウスプロバイダーまたはクラウドプロバイダーから課金されます。

の請求 AWS Clean Rooms

AWS Clean Rooms では、コラボレーションクリエイターは、コラボレーションでクエリまたはジョブのコンピューティングコストを支払うメンバーを指定できます。

ほとんどの場合、[クエリを行えるメンバー](#)と[クエリの計算コストを負担するメンバー](#)は同じです。ただし、クエリを行えるメンバーとクエリの計算コストを負担するメンバーが異なる場合は、クエリを行えるメンバーが自分のメンバーシップリソースに対してクエリを実行しても、請求は、クエリの計算コストを負担するメンバーのメンバーシップリソースに行われます。

コストを負担するメンバーは、クエリを実行しているわけでも、クエリが実行されるリソースの所有者でもないため、そのメンバーの CloudTrail イベント履歴には、実行されたクエリのイベントは表示されません。ただし、支払者には、コラボレーションでクエリを実行できるメンバーによって実行されるすべてのクエリについて、メンバーシップリソースに対して生成された料金が表示されます。

コラボレーションを作成する方法と、クエリの計算コストを負担するメンバーを設定する方法の詳細については、「[コラボレーションの作成](#)」を参照してください。

の分析ルール AWS Clean Rooms

テーブルがコラボレーション分析 AWS Clean Rooms に 使用できるようにする一環として、コラボレーションメンバーは分析ルールを設定する必要があります。

分析ルールは、各データ所有者が設定済みテーブルに設定するプライバシー強化のためのコントロールです。分析ルールによって、設定済みテーブルの分析方法が決まります。

分析ルールは、設定済みテーブルをアカウントレベルで制御するもの (アカウントレベルのリソース) で、設定済みテーブルが関連付けられているすべてのコラボレーションに適用されます。分析ルールが設定されていない場合、設定済みテーブルをコラボレーションに関連付けることはできますが、クエリは実行できません。クエリは、分析ルールの種類が同一の設定済みテーブルのみを参照できます。

分析ルールを設定するには、まず分析の種類を選択し、次に分析ルールを指定します。どちらの手順でも、有効にするユースケースと、基になるデータをどのように保護するかを検討する必要があります。

AWS Clean Rooms は、クエリで参照されるすべての設定済みテーブルに対して、より制限の厳しいコントロールを適用します。

制限に関するコントロールの例を以下に示します。

Example制限に関するコントロール: 出力の制約

- コラボレーター A の ID 列には 100 という出力の制約があります。
- コラボレーター B の ID 列には 150 という出力の制約があります。

両方の設定済みテーブルを参照する集約クエリで、クエリ出力に行を表示するには、1 つの出力行に少なくとも 150 個の個別 ID 値が必要です。クエリ出力には、出力の制約により結果が削除されたことは示されません。

Example制限に関するコントロール: 未承認の分析テンプレート

- コラボレーター A は、カスタム分析ルールで、コラボレーター A とコラボレーター B の設定済みテーブルを参照するクエリを含む分析テンプレートを許可しました。
- コラボレーター B はこの分析テンプレートを許可していません。

コラボレーター B が分析テンプレートを許可していないため、クエリを実行できるメンバーはその分析テンプレートを実行できません。

分析ルールの種類

分析ルールには、[集計](#)、[リスト](#)、[カスタム](#)の 3 種類があります。以下の表で、分析ルールの種類を比較しています。各種分析ルールの指定については、それぞれ別のセクションで説明しています。

Note

ID マッピングテーブル分析ルールと呼ばれる分析ルールタイプがあります。ただし、この分析ルールは によって管理 AWS Clean Rooms され、変更することはできません。詳細については、「[ID マッピングテーブル分析ルール](#)」を参照してください。

以下のセクションでは、各分析ルールタイプでサポートされているユースケースとコントロールについて説明します。

対応するユースケース

次の表は、各種分析ルールで対応しているユースケースの比較をまとめたものです。

ユースケース	集計	リスト	カスタム
対応する分析	COUNT、SUM、および AVG 関数を使用して、任意のディメンションで統計を集約するクエリ	複数のテーブルが重複している箇所の行レベルのリストを出力するクエリ	分析テンプレートまたは分析作成者によって確認、許可されている、あらゆるカスタム分析
一般的なユースケース	セグメント分析、測定、アトリビューション	エンリッチメント、セグメント構築	ファーストタッチアトリビューション、増分分析、オーディエンス発掘
SQL 構文	• JOIN ステートメント: INNER JOIN • 集計関数: COUNT/COUNT DISTINCT、SUM/SUM DISTINCT、および AVG • スカラー関数: 限定サブセット	• JOIN ステートメント: INNER JOIN • スカラー関数: なし	SELECT コマンドで利用できる SQL 関数と SQL 構文の大部分
サブクエリと共通テーブル式 (CTE)	いいえ	いいえ	はい
分析テンプレート	いいえ	いいえ	はい

対応するコントロール

次の表は、各分析ルールタイプで基になるデータがどのように保護されるのか、その比較をまとめたものです。

コントロール	集計	リスト	カスタム
コントロールのメカニズム	テーブル内のデータをクエリでどのように使用してよいかを制御します (hashed_email 列の COUNT と SUM を許可するなど)。	テーブル内のデータをクエリでどのように使用してよいかを制御します (hashed_email 列の使用を結合のみで許可するなど)。	テーブルでどのクエリを実行してよいかを制御します (分析テンプレート「Custom query 1」で定義されたクエリのみを許可するなど)。
組み込みのプライバシー強化機能	• ブラインドマッチ • 集約が必須 • 最小数の集約しきい値 $\geq$ • 事前定義されたクエリ構造	• ブラインドマッチ • 重複が必須 • 事前定義されたクエリ構造 • 許可される追加分析	• 差分プライバシー • 許可されない出力列
実行前のクエリの確認	いいえ	いいえ	はい。分析テンプレートを使用

で利用できる分析ルールの詳細については AWS Clean Rooms、以下のトピックを参照してください。

- [集計分析ルール](#)
- [リスト分析ルール](#)
- [のカスタム分析ルール AWS Clean Rooms](#)

集計分析ルール

では AWS Clean Rooms、集計分析ルールは、オプションのディメンションに沿って COUNT、SUM、AVG 関数を使用して集計統計を生成します。設定済みテーブルに集計分析ルールを追加すると、クエリを行えるメンバーが設定済みテーブルに対してクエリを実行できるようになります。

集計分析ルールは、キャンペーン計画、メディアリーチ、頻度測定、アトリビューションなどのユースケースに対応します。

サポートされているクエリ構造と構文は、「[集約クエリの構造と構文](#)」で定義されています。

「[集計分析ルール - クエリコントロール](#)」で定義されている分析ルールのパラメータには、クエリコントロールとクエリ結果コントロールがあります。クエリコントロールでは、直接または間接的にクエリを実行できるメンバーが所有する 1 つ以上の設定済みテーブルに、1 つの設定済みテーブルを結合することを要求できます。この要求により、クエリを自分のテーブルと相手のテーブルの交差部分 (INNERJOIN) で実行することが可能になります。

集約クエリの構造と構文

集計分析ルールが追加されたテーブルに対するクエリは、次の構文に従う必要があります。

```
--select_aggregate_function_expression
SELECT
aggregation_function(column_name) [[AS] column_alias ] [, ...]

--select_grouping_column_expression
[, {column_name|scalar_function(arguments)} [[AS] column_alias ]][, ...]

--table_expression
FROM table_name [[AS] table_alias ]
    [[INNER] JOIN table_name [[AS] table_alias] ON join_condition] [...]

--where_expression
[WHERE where_condition]

--group_by_expression
[GROUP BY {column_name|scalar_function(arguments)}, ...]]

--having_expression
[HAVING having_condition]
```

```
--order_by_expression
[ORDER BY {column_name|scalar_function(arguments)} [{ASC|DESC}]] [,...]]
```

次の表は、前述の構文で示したそれぞれの式について説明しています。

式	定義	例
<i>select_aggregate_function_expression</i>	次の式を含むカンマ区切りリストです。 - select_aggregation_function_expression - select_aggregate_expression  Note select_aggregate_expression に少なくとも 1 つの select_aggregation_function_expression が必要です。	SELECT SUM(PRICE), user_segment
<i>select_aggregation_function_expression</i>	1 つ以上の列に適用される、1 つ以上のサポートされている集約関数です。集約関数の引数として指定できるのは列だけです。	AVG(PRICE) COUNT(DISTINCT user_id)

式	定義	例
	 Note <code>select_aggregation_expression</code> には少なくとも 1 つの <code>select_aggregation_function_expression</code> が必要です。	

式	定義	例
<code>select_grouping_column_expression</code>	以下を使用する任意の式を含めることができる式です。 • テーブルの列名 • サポートされているスカラー関数 • [String literals] (文字列リテラル) • 数値リテラル	TRUNC(timestampColumn) UPPER(campaignName)

 Note

`select_aggregate_expression` では AS パラメータの有無にかかわらず、列にエイリアスを指定できます。詳細については、「[AWS Clean Rooms SQL リファレンス](#)」を参照してください。

式	定義	例
<i>table_expression</i>	join_condition で結合条件式を連結するテーブル、またはテーブルの結合。 join_condition はブール値を返します。 table_expression では、以下がサポートされています。 • 特定の JOIN 型 (INNER JOIN) • join_condition 内の等価比較条件 (=) • 論理演算子 (AND、OR)	<pre>FROM consumer_table INNER JOIN provider_ table ON consumer_table.id entifier1 = provider_ table.identifier1 AND consumer_table .identifier2 = provider_table.id entifier2</pre>

式	定義	例
<i>where_expression</i>	ブール値を返す条件式です。次のような要素で構成されています。 • テーブルの列名 • サポートされているスカラー関数 • 算術演算子 • [String literals] (文字列リテラル) • 数値リテラル サポートされている比較条件は (=, >, <, <=, >=, <>, !=, NOT, IN, NOT IN, LIKE, IS NULL, IS NOT NULL) です。 サポートされている論理演算子は (AND, OR) です。 where_expression はオプションです。	<pre>WHERE where_condition</pre> <pre>WHERE price > 100</pre> <pre>WHERE TRUNC(timestampColumn) = '1/1/2022'</pre> <pre>WHERE timestampColumn = timestampColumn2 - 14</pre>
<i>group_by_expression</i>	select_grouping_column_expression の要件に合致する式のカンマ区切りリストです。	<pre>GROUP BY TRUNC(timestampColumn), UPPER(campaignName), segment</pre>

式	定義	例
<i>having_expression</i>	ブール値を返す条件式です。サポートされている集約関数が 1 つの列に適用され (例えば SUM(price))、数値リテラルと比較されます。 サポートされている条件は (=, >, <, <=, >=, <>, !=) です。 サポートされている論理演算子は (AND, OR) です。 having_expression はオプションです。	HAVING SUM(SALES) > 500

式	定義	例
<code>order_by_expression</code>	前述の <code>select_aggregate_expression</code> で定義されているものと同じ要件と互換性のある式のカンマ区切りリストです。 <code>order_by_expression</code> はオプションです。  Note <code>order_by_expression</code> では ASC と DESC のパラメータを使用できません。詳細については、「AWS Clean Rooms SQL リファレンス」で ASC DESC パラメータを参照してください。	<code>ORDER BY SUM(SALES), UPPER(campaignName)</code>

集約クエリの構造と構文については、次の点に注意してください。

- SELECT 以外の SQL コマンドはサポートされていません。
- サブクエリと共通テーブル式 (WITH など) はサポートされていません。
- 複数のクエリを組み合わせる演算子 (UNION など) はサポートされていません。
- TOP、LIMIT、および OFFSET パラメータはサポートされていません。

集計分析ルール - クエリコントロール

集約クエリコントロールを使用すると、テーブル内の列を使用してテーブルにクエリを実行する方法を制御できます。例えば、どの列を結合に使用するか、どの列がカウントされるようにするか、WHERE ステートメントでどの列を使用できるようにするかを制御できます。

以下のセクションでは、それぞれのコントロールについて説明します。

トピック

- [集約コントロール](#)
- [結合コントロール](#)
- [ディメンションコントロール](#)
- [スカラー関数](#)

集約コントロール

集約コントロールを使用すると、どの集約関数を許可し、どの列に適用するかを定義できます。集約関数は、SELECT、HAVING、および ORDER BY の式で使用できます。

コントロール	定義	使用方法
aggregateColumns	集約関数での使用を許可する設定済みテーブル列の列。	aggregateColumns は、SELECT、HAVING、および ORDER BY の式の集約関数で使用できます。 一部の aggregateColumns は joinColumn としても分類されることがあります (後述)。 特定の aggregateColumn は dimensionColumn としても分類されることはありません (後述)。

コントロール	定義	使用方法
function	aggregateColumns で使用できる COUNT 関数、SUM 関数、および AVG 関数。	function は関連付けられている aggregateColumns に適用できます。

結合コントロール

JOIN 句を使用して、2 つ以上のテーブルの行を、テーブル間の関連する列に基づいて結合します。

結合コントロールを使用して、テーブルを の他のテーブルに結合する方法を制御できます。table_expression。 は INNER AWS Clean Rooms のみをサポートします。JOIN。 INNERJOIN ステートメントは、定義したコントロールに従って、分析ルールjoinColumnで として明示的に分類された列のみを使用できます。

INNER JOIN は、自身の設定済みテーブルの joinColumn と、コラボレーション内のもう 1 つの設定済みテーブルの joinColumn で動作する必要があります。テーブルのどの列を joinColumn として使用できるようにするかはテーブルの所有者が決定します。

ON 句内の一致条件ごとに、2 つの列間の等価比較条件 (=) を使用する必要があります。

ON 句では次のようにして複数の一致条件を使用できます。

- AND 論理演算子を使用して組み合わせる
- OR 論理演算子を使用して区切る

Note

JOIN の一致条件では、必ず JOIN の各側の 1 つの行が一致しなければなりません。OR または AND 論理演算子で接続されるすべての条件がこの要件を満たす必要があります。

AND 論理演算子を使用したクエリの例を以下に示します。

```
SELECT some_col, other_col
FROM table1
JOIN table2
ON table1.id = table2.id AND table1.name = table2.name
```

OR 論理演算子を使用したクエリの例を以下に示します。

```
SELECT some_col, other_col
FROM table1
  JOIN table2
  ON table1.id = table2.id OR table1.name = table2.name
```

コントロール	定義	使用方法
joinColumns	クエリを行えるメンバーに INNER JOIN ステートメントでの使用を許可する列 (ある場合)。	特定の joinColumn は aggregateColumn としても分類されることがあります (「集約コントロール」を参照)。 同じ列を joinColumn と dimensionColumns の両方として使用することはできません (後述)。 aggregateColumn としても分類されていない限り、INNER JOIN 以外のクエリの他のどの部分でも joinColumn を使用することはできません。
joinRequired	クエリを行えるメンバーの設定済みテーブルとの INNER JOIN を必須にするかどうかを制御します。	このパラメータを有効した場合、INNER JOIN は必須になります。このパラメータを有効にしない場合、INNER JOIN はオプションになります。 このパラメータを有効にすると、クエリを行えるメンバーは、自身が所有するテーブルを INNER JOIN に必ず含めなければなりません。クエリ

コントロール	定義	使用方法
		を行えるメンバーは、自身のテーブルを相手のテーブルと直接または間接的 (つまり自分のテーブルを、相手のテーブルと結合されている別のテーブルに結合) に JOIN する必要があります。

以下は間接的な結合の例です。

```
ON
my_table.identifer = third_party_table.identifier
....
ON
third_party_table.identifier = member_who_can_query_table.id
```

Note

クエリを行えるメンバーが、joinRequired パラメータを使用することもできます。その場合、クエリで自分のテーブルを少なくとも 1 つの他のテーブルと結合する必要があります。

ディメンションコントロール

ディメンションコントロールは、集約列をフィルタリング、グループ化、または集計する際の基準となる列を制御します。

コントロール	定義	使用方法
dimensionColumns	クエリを行えるメンバーに SELECT、WHERE、GROUP、BY、および ORDER BY の使用を許可する列 (ある場合)。	dimensionColumn は、SELECT (select_grouping_column_expression)、WHERE、GROUP、BY、および ORDER BY で使用できます。

コントロール	定義	使用方法
		同じ列を dimension Column と joinColumn または aggregateColumn の両方として使用することはできません。

スカラー関数

スカラー関数は、どのスカラー関数をディメンション列に使用できるかを制御します。

コントロール	定義	使用方法
scalarFunctions	クエリの dimension Columns で使用できるスカラー関数。	dimensionColumns での適用を許可するスカラー関数 (CAST など) を指定します (ある場合)。 スカラー関数を他の関数と重ねて使用したり、他の関数内で使用したりすることはできません。スカラー関数の引数には、列、文字列リテラル、または数値リテラルを使用できます。

以下のスカラー関数がサポートされています。

- 数学関数 – ABS、CEILING、FLOOR、LOG、LN、ROUND、SQRT
- データ型フォーマット関数 – CAST, CONVERT, TO_CHAR, TO_DATE, TO_NUMBER, TO_TIMESTAMP
- 文字列関数 – LOWER、UPPER、TRIM、RTRIM、SUBSTRING
 - RTRIM では、カスタム文字セットをトリミングすることはできません。
- 条件式 – COALESCE
- 日付関数 – EXTRACT、GETDATE、CURRENT_DATE、DATEADD

• その他の関数 – TRUNC

詳細については、「[AWS Clean Rooms SQL リファレンス](#)」を参照してください。

集計分析ルール - クエリ結果コントロール

集約クエリ結果コントロールでは、返される各出力行が満たす必要がある条件を 1 つ以上指定することで、どの結果を返すかを制御できます。AWS Clean Rooms では、COUNT (DISTINCT column) >= X という形式の集約制約をサポートしています。この形式では、設定済みテーブルから、選択した少なくとも X 個の個別値 (例えば、個別の user_id 値の最小数) を各行が集約することになります。送信されたクエリ自体が指定された列を使用しない場合でも、この最小数のしきい値は自動的に適用されます。これらは、コラボレーション内の各メンバーの設定済みテーブルから、クエリ内の各設定済みテーブルにまとめて適用されます。

各設定済みテーブルの分析ルールには、少なくとも 1 つの集約制約が必要です。設定済みテーブルの所有者が複数の columnName や関連付けられた minimum を追加すると、それらはまとめて適用されます。

集約制約

集約制約は、クエリ結果のどの行を返すかを制御します。行が返されるには、その行が、集約制約で指定された各列で、指定された個別値の最小数を満たす必要があります。この要件は、クエリや分析ルールの他の部分にその列が明示的に記述されていない場合でも適用されます。

コントロール	定義	使用方法
columnName	各出力行が満たす必要がある条件で使用される aggregate Column です。	設定済みテーブル内のどの列でもかまいません。
minimum	クエリ結果で返される出力行に含まれていなければならない、関連付けられた aggregateColumn の個別値の最小数です (COUNT DISTINCT など)。	minimum の値は 2 以上にする必要があります。

集計分析ルールの構造

次の例は、集計分析ルールの事前定義された構造を示しています。

次の例では、*MyTable* がデータテーブルを表しています。各#####は、独自の情報に置き換えることができます。

```
{
  "aggregateColumns": [
    {
      "columnNames": [MyTable column names], "function": [Allowed Agg Functions]
    },
  ],
  "joinRequired": ["QUERY_RUNNER"],
  "joinColumns": [MyTable column names],
  "dimensionColumns": [MyTable column names],
  "scalarFunctions": [Allowed Scalar functions],
  "outputConstraints": [
    {
      "columnName": [MyTable column names], "minimum": [Numeric value]
    },
  ],
}
```

集計分析ルール - 例

次の例は、2つの企業が集約分析 AWS Clean Rooms を使用してコラボレーションする方法を示しています。

A 社には顧客データと売上データがあります。A 社は製品の返品状況を把握したいと考えています。B 社は A 社の小売業者の一社で、返品データを保有しています。B 社には、A 社にとって有益な、顧客に関するセグメント属性 (関連製品を購入した、小売業者からカスタマーサービスを利用したなど) のデータもあります。B 社は、行レベルの顧客返品データや属性情報は提供したくありません。B 社の希望は、重複する顧客に関する統計情報を最小数の集約しきい値で A 社が取得できるように、一連のクエリを有効にすることです。

A 社が製品の返品状況を把握し、B 社やその他のチャネルでより良い製品を提供できるように、A 社と B 社は協力することにしました。

コラボレーションを作成して集計分析を行うために、両社は次のことを行います。

1. A 社がコラボレーションを作成し、メンバーシップを作成します。このコラボレーションには、B 社がコラボレーションの相手方メンバーとして参加します。A 社はコラボレーションでのクエリログ記録を有効にし、自社アカウントでのクエリログの記録を有効にします。
2. B 社がコラボレーションでメンバーシップを作成し、そのアカウントでのクエリログの記録を有効にします。
3. A 社が、設定済み売上テーブルを作成します。
4. A 社が、設定済み売上テーブルに次の集計分析ルールを追加します。

```
{
  "aggregateColumns": [
    {
      "columnNames": [
        "identifier"
      ],
      "function": "COUNT_DISTINCT"
    },
    {
      "columnNames": [
        "purchases"
      ],
      "function": "AVG"
    },
    {
      "columnNames": [
        "purchases"
      ],
      "function": "SUM"
    }
  ],
  "joinColumns": [
    "hashedemail"
  ],
  "dimensionColumns": [
    "demoseg",
    "purchasedate",
    "productline"
  ],
  "scalarFunctions": [
    "CAST",
    "COALESCE",
    "TRUNC"
  ],
}
```

```
"outputConstraints": [  
  {  
    "columnName": "hashedemail",  
    "minimum": 2,  
    "type": "COUNT_DISTINCT"  
  },  
]  
}
```

aggregateColumns – A 社は、売上データと返品データで重複している一意の顧客の数をカウントしたいと考えています。また A 社は、purchases の数を合計して、returns の数と比較したいと考えています。

joinColumns – A 社は、identifier を使用して売上データの顧客と返品データの顧客を照合したいと考えています。これにより、A 社は返品を適切な購入と照合できるようになります。また、A 社が重複する顧客をセグメント化するのにも役立ちます。

dimensionColumns – A 社は、dimensionColumns を使用して、特定の製品での絞り込みを行い、一定期間にわたって購入と返品を比較して返品日が購入日付より後になるものを確認し、重複する顧客をセグメント化します。

scalarFunctions – A 社は、A 社がコラボレーションに関連付けた設定済みテーブルに基づいて、必要に応じてデータ型フォーマットを更新できるよう、CAST スカラー関数を選択します。また、必要に応じて列のフォーマットに役立つスカラー関数も追加します。

outputConstraints – A 社は最小数の出力制約を設定します。アナリストは売上テーブルから行レベルのデータを確認できるため、結果を制約する必要はありません。

Note

企業 A は分析ルールに joinRequired を追加しません。これにより、アナリストは売上テーブルだけに対して柔軟にクエリを実行できます。

5. B 社が、設定済み返品テーブルを作成します。
6. B 社が、設定済み返品テーブルに次の集計分析ルールを追加します。

```
{  
  "aggregateColumns": [  
    {  
      "columnNames": [  

```

```
    "identifier"
  ],
  "function": "COUNT_DISTINCT"
},
{
  "columnNames": [
    "returns"
  ],
  "function": "AVG"
},
{
  "columnNames": [
    "returns"
  ],
  "function": "SUM"
}
],
"joinColumns": [
  "hashedemail"
],
"joinRequired": [
  "QUERY_RUNNER"
],
"dimensionColumns": [
  "state",
  "popularpurchases",
  "customerserviceuser",
  "productline",
  "returndate"
],
"scalarFunctions": [
  "CAST",
  "LOWER",
  "UPPER",
  "TRUNC"
],
"outputConstraints": [
  {
    "columnName": "hashedemail",
    "minimum": 100,
    "type": "COUNT_DISTINCT"
  },
  {
    "columnName": "producttype",
```

```

        "minimum": 2,
        "type": "COUNT_DISTINCT"
    }
]
}

```

aggregateColumns – B 社は、A 社が `returns` の数を合計して購入数と比較できるようにします。集約クエリを有効にしているため、少なくとも 1 つは集約列があります。

joinColumns – B 社は、A 社が返品データの顧客と売上データの顧客を照合できるよう、`identifier` での結合を有効にします。`identifier` データは特に機密性が高く、`joinColumn` として指定すれば、データがクエリで出力されることはありません。

joinRequired – B 社は、返品データと売上データの重複に関するクエリを必須にします。A 社が B 社のデータセット内のすべての個人を照会できるようにはしたくありません。その制限についてはコラボレーション契約でも合意しています。

dimensionColumns – B 社は、A 社が `state`、`popularpurchases`、`customerserviceuser` の属性でフィルタとグループ化を実行できるようにします。これらは A 社の分析に役立つ固有の属性です。B 社は、A 社が `returndate` を使用して `purchasedate` の後に発生した `returndate` で出力をフィルタリングできるようにします。このフィルタリングにより、出力がより正確になり、製品変更の影響を評価できるようになります。

scalarFunctions – B 社は以下を有効にします。

- 日付の `TRUNC`
- `producttype` が異なる形式でデータに入力された場合の `LOWER` と `UPPER`
- A 社が売上のデータ型を返品のデータ型と同じものに変換する必要がある場合の `CAST`

A 社は、他のスカラー関数はクエリに必要ではないと考えているため、有効にしません。

outputConstraints – B 社は、顧客の再識別が困難になるように、`hashedemail` に最小数の出力制約を設定します。また、返品された特定の製品の再識別が困難になるように、`producttype` にも最小数の出力制約を設定します。出力のディメンション (`state` など) によっては、特定の製品タイプがより優勢になる可能性があります。A 社がデータに追加した出力の制約にかかわらず、B 社の出力の制約が常に適用されます。

7. A 社が、売上テーブルとコラボレーションとの関連付けを作成します。

8. B 社が、返品テーブルとコラボレーションとの関連付けを作成します。

9. A 社が、B 社の返品数を 2022 年の場所別の購入総数と比較して詳しく把握するために、次の例のようなクエリを実行します。

```
SELECT
    companyB.state,
    SUM(companyB.returns),
    COUNT(DISTINCT companyA.hashemail)
FROM
    sales companyA
    INNER JOIN returns companyB ON companyA.identifier = companyB.identifier
WHERE
    companyA.purchasedate BETWEEN '2022-01-01' AND '2022-12-31' AND
    TRUNC(companyB.returndate) > companyA.purchasedate
GROUP BY
    companyB.state;
```

10A 社と B 社がクエリログを確認します。B 社は、クエリがコラボレーション契約で合意された内容と一致していることを確認します。

集計分析ルールに関する問題のトラブルシューティング

集計分析ルール使用時の一般的な問題については、ここにある情報を使用して診断と修正を行ってください。

問題

- [クエリから何も結果が返されない](#)

クエリから何も結果が返されない

この問題は、一致する結果がない場合や、一致する結果が最小数の集約しきい値を 1 つ以上満たしていない場合に発生する可能性があります。

最小数の集約しきい値の詳細については、「[集計分析ルール - 例](#)」を参照してください。

リスト分析ルール

では AWS Clean Rooms、リスト分析ルールは、追加された設定済みテーブルとクエリ可能なメンバーの設定済みテーブルとの重複の行レベルのリストを出力します。クエリを行えるメンバーが、リスト分析ルールを含むクエリを実行します。

リスト分析ルールは、エンリッチメントやオーディエンス構築などのユースケースに対応します。

この分析ルール of 事前定義されたクエリ構造と構文の詳細については、「[リスト分析ルールの事前定義された構造](#)」を参照してください。

[リスト分析ルール - クエリコントロール](#) で定義されているリスト分析ルールのパラメータにはクエリコントロールがあります。クエリコントロールには、出力に表示できる列を選択する機能が含まれています。クエリには、直接または間接的にクエリを実行できるメンバーの設定済みテーブルとの結合が少なくとも 1 つ必要です。

[集計分析ルール](#) のようなクエリ結果コントロールはありません。

リストクエリでは算術演算子しか使用できません。他の関数 (集約やスカラーなど) は使用できません。

トピック

- [リストクエリの構造と構文](#)
- [リスト分析ルール - クエリコントロール](#)
- [リスト分析ルールの事前定義された構造](#)
- [リスト分析ルール - 例](#)

リストクエリの構造と構文

リスト分析ルールが追加されたテーブルに対するクエリは、次の構文に従う必要があります。

```
--select_list_expression
SELECT DISTINCT column_name [[AS] column_alias ] [, ...]

--table_expression
FROM table_name [[AS] table_alias ]
    [[INNER] JOIN table_name [[AS] table_alias] ON join_condition] [...]

--where_expression
[WHERE where_condition]

--limit_expression
[LIMIT number]
```

次の表は、前述の構文で示したそれぞれの式について説明しています。

式	定義	例
<code>select_list_expression</code>	テーブル列名を 1 つ以上含むカンマ区切りリストです。 DISTINCT パラメータが必須です。  Note <code>select_list_expression</code> では AS パラメータの有無にかかわらず、列にエイリアスを指定できます。 詳細については、「AWS Clean Rooms SQL リファレンス」を参照してください。	<pre>SELECT DISTINCT segment</pre>
<code>table_expression</code>	テーブルまたはテーブルの結合と、それを <code>join_condition</code> に連結するための <code>join_condition</code>。 <code>join_condition</code> はブール値を返します。 <code>table_expression</code> では、以下がサポートされています。 - 特定の JOIN 型 (INNER JOIN) <code>join_condition</code> 内の等価比較条件 (=)	<pre>FROM consumer_table INNER JOIN provider_table ON consumer_table.identifier1 = provider_table.identifier1 AND consumer_table.identifier2 = provider_table.identifier2</pre>

式	定義	例
	論理演算子 (AND、OR)	
<i>where_expression</i>	ブール値を返す条件式です。次のような要素で構成されています。 - テーブルの列名 - 算術演算子 [String literals] (文字列リテラル) - 数値リテラル サポートされている比較条件は (=, >, <, <=, >=, <>, !=, NOT, IN, NOT IN, LIKE, IS NULL, IS NOT NULL) です。 サポートされている論理演算子は (AND, OR) です。 where_expression はオプションです。	<pre>WHERE state + '_' + city = 'NY_NYC'</pre> <pre>WHERE timestampColumn = timestampColumn2 - 14</pre>
<i>limit_expression</i>	この式は正の整数にする必要があります。 limit_expression はオプションです。	<pre>LIMIT 100</pre>

リストクエリの構造と構文については、次の点に注意してください。

- SELECT 以外の SQL コマンドはサポートされていません。
- サブクエリと共通テーブル式 (WITH など) はサポートされていません。
- HAVING、GROUP BY、および ORDER BY 句はサポートされていません。

- OFFSET パラメータはサポートされていません。

リスト分析ルール - クエリコントロール

リストクエリコントロールを使用すると、テーブル内の列を使用してテーブルにクエリを実行する方法を制御できます。例えば、どの列を結合に使用するか、SELECT ステートメントや WHERE 句でどの列を使用できるようにするかを制御できます。

以下のセクションでは、それぞれのコントロールについて説明します。

トピック

- [結合コントロール](#)
- [リストコントロール](#)

結合コントロール

結合コントロールを使用することで、テーブルを `table_expression` で他のテーブルに結合する方法を制御できます。AWS Clean Rooms は、INNER JOIN のみをサポートしています。リスト分析ルールでは、少なくとも 1 つの INNER JOIN が必要であり、クエリを行えるメンバーは、自身が所有するテーブルを INNER JOIN に必ず含めなければなりません。つまり、クエリを行えるメンバーは、自身のテーブルを相手のテーブルと直接または間接的に結合する必要があります。

以下は間接的な結合の例です。

```
ON
my_table.identifer = third_party_table.identifier
....
ON
third_party_table.identifier = member_who_can_query_table.id
```

INNER JOIN ステートメントでは、分析ルールで明示的に `joinColumn` として分類された列のみを使用できます。

INNER JOIN は、自身の設定済みテーブルの `joinColumn` と、コラボレーション内のもう 1 つの設定済みテーブルの `joinColumn` で動作する必要があります。テーブルのどの列を `joinColumn` として使用できるようにするかはテーブルの所有者が決定します。

ON 句内の一致条件ごとに、2 つの列間の等価比較条件 (=) を使用する必要があります。

ON 句では次のようにして複数の一致条件を使用できます。

- AND 論理演算子を使用して組み合わせる
- OR 論理演算子を使用して区切る

 Note

JOIN の一致条件では、必ず JOIN の各側の 1 つの行が一致しなければなりません。OR または AND 論理演算子で接続されるすべての条件がこの要件を満たす必要があります。

AND 論理演算子を使用したクエリの例を以下に示します。

```
SELECT some_col, other_col
FROM table1
  JOIN table2
  ON table1.id = table2.id AND table1.name = table2.name
```

OR 論理演算子を使用したクエリの例を以下に示します。

```
SELECT some_col, other_col
FROM table1
  JOIN table2
  ON table1.id = table2.id OR table1.name = table2.name
```

コントロール	定義	使用方法
joinColumns	クエリを行えるメンバーに INNER JOIN ステートメントでの使用を許可する列。	同じ列を joinColumn と listColumn の両方に分類することはできません (「 リストコントロール 」を参照)。 joinColumn は INNER JOIN 以外のクエリのどの部分でも使用できません。

リストコントロール

リストコントロールは、クエリ出力に一覧表示できる (SELECT ステートメントで使用される) 列や、結果のフィルタリングに使用できる (WHERE ステートメントで使用される) 列を制御します。

コントロール	定義	使用方法
listColumns	クエリを行えるメンバーに SELECT および WHERE で使用を許可する列。	listColumn は SELECT と WHERE で使用できます。 同じ列を listColumn と joinColumn の両方として使用することはできません。

リスト分析ルールの事前定義された構造

次の例には、リスト分析ルールを完成させる方法を示す事前定義された構造が含まれています。

次の例では、*MyTable* がデータテーブルを表しています。各#####は、独自の情報に置き換えることができます。

```
{
  "joinColumns": [MyTable column name(s)],
  "listColumns": [MyTable column name(s)],
}
```

リスト分析ルール - 例

次の例は、2 つの企業がリスト分析 AWS Clean Rooms を使用してコラボレーションする方法を示しています。

A 社には顧客関係管理 (CRM) データがあります。A 社は、顧客についてさらに詳しく知るため、また属性を他の分析のインプットとして使用するために、顧客に関する追加のセグメントデータを取得したいと考えています。B 社には、自社で入手したデータに基づいて作成した独自のセグメント属性で構成されるセグメントデータがあります。B 社は、自社のデータと A 社のデータの間に重複している顧客についてのみ、固有のセグメント属性を A 社に提供したいと考えています。

両社は、A 社が重複するデータのエンリッチメントを行えるよう協力することにしました。A 社はクエリを行えるメンバーで、B 社はデータを寄稿するメンバーです。

コラボレーションを作成し、コラボレーションでリスト分析を実行するために、各社は次のことを行います。

1. A 社がコラボレーションを作成し、メンバーシップを作成します。このコラボレーションには、B 社がコラボレーションの相手方のメンバーとして参加します。A 社はコラボレーションでのクエリログ記録を有効にし、自社アカウントでのクエリログの記録を有効にします。
2. B 社がコラボレーションでメンバーシップを作成し、そのアカウントでのクエリログの記録を有効にします。
3. A 社が、設定済み CRM テーブルを作成します。
4. A 社が、次の例のような分析ルールを設定済み顧客テーブルに追加します。

```
{
  "joinColumns": [
    "identifier1",
    "identifier2"
  ],
  "listColumns": [
    "internalid",
    "segment1",
    "segment2",
    "customercategory"
  ]
}
```

joinColumns – A 社は、hashedemail と thirdpartyid (ID ベンダーから取得) を使用して CRM データの顧客とセグメントデータの顧客を照合したいと考えています。これにより、A 社は適切な顧客のエンリッチメントデータと照合できるようになります。JoinColumns が 2 つあることで、分析の一致率が向上する可能性があります。

listColumns – A 社は、listColumns を使用して、自社のシステム内で使用している internalid に加えてエンリッチメントのための列を取得します。segment1、segment2、customercategory を追加し、フィルタで使用することでエンリッチメントを特定のセグメントに限定できるようにします。

5. B 社が、設定済みセグメントテーブルを作成します。
6. B 社が、分析ルールを設定済みセグメントテーブルに追加します。

```
{
  "joinColumns": [
```

```
    "identifier2"
  ],
  "listColumns": [
    "segment3",
    "segment4"
  ]
}
```

joinColumns – B 社は、A 社が `identifier2` で結合を実行して、セグメントデータの顧客を CRM データの顧客と照合できるようにします。A 社と B 社は ID ベンダーと協力して、今回のコラボレーションに適していそうな `identifier2` を取得しています。`identifier2` は最も一致率が高く、最も正確で、これ以外の ID はクエリに必要なと考えられたため、他の `joinColumns` は追加しませんでした。

listColumns – B 社は、自社で作成、収集し、データエンリッチメントに含めるために (A 社とともに) 調整した属性である `segment3` および `segment4` の属性で、A 社がエンリッチメントを行えるようにします。こうした重複するセグメントを A 社に行レベルで取得してもらうのは、これがデータエンリッチメントのコラボレーションであるためです。

7. A 社が、CRM テーブルとコラボレーションとの関連付けを作成します。
8. B 社が、セグメントテーブルとコラボレーションとの関連付けを作成します。
9. A 社が、次のようなクエリを実行して、重複する顧客データのエンリッチメントを行います。

```
SELECT companyA.internalid, companyB.segment3, companyB.segment4
INNER JOIN returns companyB
  ON companyA.identifier2 = companyB.identifier2
WHERE companyA.customercategory > 'xxx'
```

- 10 A 社と B 社がクエリログを確認します。B 社は、クエリがコラボレーション契約で合意された内容と一致していることを確認します。

のカスタム分析ルール AWS Clean Rooms

では AWS Clean Rooms、カスタム分析ルールは、設定されたテーブルでカスタムクエリを実行できるようにする新しいタイプの分析ルールです。カスタム SQL クエリは、現在も `SELECT` コマンドの使用のみに制限されていますが、[集約クエリ](#)や[リストクエリ](#)よりも多くの SQL 構文を使用できます (例えば、ウィンドウ関数、`OUTER JOIN`、`CTE`、サブクエリなど。詳細なリストについては、「[AWS Clean Rooms SQL リファレンス](#)」を参照してください)。カスタム SQL クエリでは、[集約クエリ](#)や[リストクエリ](#)のようなクエリ構造に従う必要はありません。

カスタム分析ルールは、カスタムアトリビューション分析、ベンチマーキング、増分解析、オーディエンス発掘など、集計分析ルールやリスト分析ルールよりも高度なユースケースに対応します。これは、集計分析ルールおよびリスト分析ルールでサポートされるユースケースのスーパーセットに追加されるものです。

カスタム分析ルールは差分プライバシーもサポートします。差分プライバシーは、データプライバシー保護のための数学的に厳格なフレームワークです。詳細については、「[AWS Clean Rooms 差分プライバシー](#)」を参照してください。分析テンプレートを作成すると、AWS Clean Rooms 差分プライバシーはテンプレートをチェックして、AWS Clean Rooms 差分プライバシーの汎用クエリ構造と互換性があるかどうかを確認します。この検証により、差分プライバシー保護テーブルでは許可されない分析テンプレートが作成されなくなります。

カスタム分析ルールを設定する場合、データ所有者は、[分析テンプレート](#)に保存されている特定のカスタムクエリを自身の設定済みテーブルで実行することを許可できます。データ所有者は、分析テンプレートを確認してから、カスタム分析ルールの許可された分析コントロールに追加します。分析テンプレートは、(テーブルが他のコラボレーションに関連付けられている場合でも) 作成されたコラボレーションでのみ使用および表示され、そのコラボレーションでクエリを行えるメンバーのみが実行できます。

または、他のメンバー (クエリプロバイダー) が確認なしでクエリを作成できるよう許可することもできます。その場合は、許可対象のクエリプロバイダーが管理するクエリプロバイダーのアカウントをカスタム分析ルールに追加します。クエリプロバイダーがクエリを行えるメンバーであれば、設定済みテーブルで任意のクエリを直接実行できます。クエリプロバイダーは、[分析テンプレートを作成](#)することによってクエリを作成することもできます。クエリプロバイダーによって作成されたクエリは、が存在し、テーブル AWS アカウント が関連付けられているすべてのコラボレーションで、テーブルで自動的に実行できます。

データ所有者がクエリの作成を許可できるのは分析テンプレートまたはアカウントのみで、両方には許可できません。データ所有者がこれを空欄のままにすると、クエリを行えるメンバーは設定済みテーブルでクエリを実行できません。

トピック

- [カスタム分析ルールの事前定義された構造](#)
- [カスタム分析ルールの例](#)
- [差分プライバシーによるカスタム分析ルール](#)

カスタム分析ルールの事前定義された構造

次の例には、差分プライバシーが有効なカスタム分析ルールを完成させる方法を示す、事前定義された構造が含まれています。userIdentifier 値は user_id のような、ユーザーを一意に識別する列です。コラボレーションで差分プライバシーが有効になっているテーブルが 2 つ以上ある場合は、AWS Clean Rooms では、テーブル間でユーザーの一貫した定義を維持するために、両方の分析ルールで同じ列をユーザー ID 列として設定する必要があります。

```
{
  "allowedAnalyses": ["ANY_QUERY"] | string[],
  "allowedAnalysisProviders": [],
  "differentialPrivacy": {
    "columns": [
      {
        "name": "userIdentifier"
      }
    ]
  }
}
```

次のいずれかを行うことができます。

- 分析テンプレート ARN を許可された分析コントロールに追加します。この場合、allowedAnalysisProviders コントロールは含まれません。

```
{
  allowedAnalyses: string[]
}
```

- AWS アカウント IDs を allowedAnalysisProviders コントロールに追加します。この場合は、ANY_QUERY を allowedAnalyses コントロールに追加します。

```
{
  allowedAnalyses: ["ANY_QUERY"],
  allowedAnalysisProviders: string[]
}
```

カスタム分析ルールの例

次の例は、2つの企業がカスタム分析ルール AWS Clean Rooms を使用して コラボレーションする方法を示しています。

A 社には顧客データと売上データがあります。A 社は、B 社のサイトで実施した広告キャンペーンによる売上の増分を把握したいと考えています。B 社には、A 社にとって有用な閲覧データとセグメント属性 (広告を閲覧する際に使用したデバイスなど) があります。

企業 A には、コラボレーションで実行したい特定の増分クエリがあります。

コラボレーションを作成し、コラボレーションでカスタム分析を実行するために、各社は次のことを行います。

1. A 社がコラボレーションを作成し、メンバーシップを作成します。このコラボレーションには、B 社がコラボレーションの相手方のメンバーとして参加します。A 社はコラボレーションでのクエリログ記録を有効にし、自社アカウントでのクエリログの記録を有効にします。
2. B 社がコラボレーションでメンバーシップを作成し、そのアカウントでのクエリログの記録を有効にします。
3. A 社が、設定済み CRM テーブルを作成します。
4. A 社が、設定済み売上テーブルに空のカスタム分析ルールを追加します。
5. A 社が、設定済み売上テーブルをコラボレーションに関連付けます。
6. B 社が、設定済み閲覧者数テーブルを作成します。
7. B 社が、設定済み閲覧者数テーブルに空のカスタム分析ルールを追加します。
8. B 社が、設定済み閲覧者数テーブルをコラボレーションに関連付けます。
9. A 社が、コラボレーションに関連付けられている売上テーブルと閲覧者数テーブルを表示し、キャンペーン月の増分クエリとパラメータを追加して分析テンプレートを作成します。

```
{
  "analysisParameters": [
    {
      "defaultValue": ""
      "type": "DATE"
      "name": "campaign_month"
    }
  ],
  "description": "Monthly incrementality query using sales and viewership data"
  "format": "SQL"
  "name": "Incrementality analysis"
```

```

"source":
  "WITH labeleddata AS
  (
    SELECT hashedemail, deviceid, purchases, unitprice, purchasedate,
    CASE
      WHEN testvalue IN ('value1', 'value2', 'value3') THEN 0
      ELSE 1
    END AS testgroup
    FROM viewershipdata
  )
  SELECT labeleddata.purchases, provider.impressions
  FROM labeleddata
  INNER JOIN salesdata
    ON labeleddata.hashedemail = provider.hashedemail
  WHERE MONTH(labeleddata.purchasedate) > :campaignmonth
  AND testgroup = :group
"
}

```

10A 社が、自社のアカウント (444455556666 など) をカスタム分析ルール of 許可された分析プロバイダーコントロールに追加します。A 社は、作成したすべてのクエリを設定済み販売テーブルで実行できるようにしたいため、許可された分析プロバイダーコントロールを使用しています。

```

{
  "allowedAnalyses": [
    "ANY_QUERY"
  ],
  "allowedAnalysisProviders": [
    "444455556666"
  ]
}

```

11B 社が、作成された分析テンプレートをコラボレーション内で確認し、クエリ文字列やパラメータなどの内容を確認します。

12B 社が、分析テンプレートが増分のユースケースに対応しており、設定済み視聴者数テーブルのクエリの実行方法がプライバシー要件を満たしていることを判断します。

13B 社が、視聴者数テーブルのカスタム分析ルール of 許可された分析コントロールに分析テンプレート ARN を追加します。B 社は、設定済み視聴者数テーブルでのみ増分クエリを実行できるようにしたいため、許可された分析コントロールを使用しています。

```

{

```

```
"allowedAnalyses": [  
  "arn:aws:cleanrooms:us-east-1:111122223333:membership/41327cc4-bbf0-43f1-b70c-a160dddceb08/analysistemplate/1ff1bf9d-781c-418d-a6ac-2b80c09d6292"  
]  
}
```

14A 社が分析テンプレートを実行し、パラメータ値 05-01-2023 を使用します。

差分プライバシーによるカスタム分析ルール

では AWS Clean Rooms、カスタム分析ルールは差分プライバシーをサポートしています。差分プライバシーは、データプライバシー保護のための数学的に厳格なフレームワークであり、データを再識別されないように保護するのに役立ちます。

差分プライバシーは、広告キャンペーン計画、広告キャンペーン後の測定、金融機関コンソーシアムにおけるベンチマーク、医療研究のための A/B テストなどの集計分析をサポートします。

サポートされているクエリ構造と構文は、「[クエリの構造と構文](#)」で定義されています。

差分プライバシーを使用したカスタム分析ルールの例

Note

AWS Clean Rooms 差分プライバシーは、データが Amazon S3 に保存されているコラボレーションでのみ使用できます。

前のセクションで説明した[カスタム分析ルールの例](#)を考えてみます。この例は、差分プライバシーを使用して再識別の試みからデータを保護すると同時に、パートナーがデータからビジネスクリティカルなインサイトを学べるようにする方法を示しています。閲覧者データを保有している B 社が、差分プライバシーを使用してデータを保護したいと想定します。差分プライバシーの設定を完了するために、B 社は次のステップを実行します。

1. B 社は差分プライバシーを有効にし、設定済み閲覧者数テーブルにカスタム分析ルールを追加します。B 社は viewershipdata.hashemail をユーザー識別子列として選択します。
2. B 社はコラボレーションに[差分プライバシーポリシーを追加し](#)、閲覧者データテーブルをクエリに使用できるようにします。B 社はデフォルトポリシーを選択し、設定を迅速に完了します。

A社は、B社のサイトでの広告キャンペーンの売上増分を把握したいと考えており、分析テンプレートを実行します。クエリは AWS Clean Rooms Differential Privacy の汎用[クエリ構造](#)と互換性があるため、クエリは正常に実行されます。

クエリの構造と構文

差分プライバシーが有効になっているテーブルが 1 つ以上含まれているクエリは、次の構文に従う必要があります。

```
query_statement:
    [cte, ...] final_select

cte:
    WITH sub_query AS (
        inner_select
        [ UNION | INTERSECT | UNION_ALL | EXCEPT/MINUS ]
        [ inner_select ]
    )

inner_select:
    SELECT [user_id_column, ] expression [, ...]
    FROM table_reference [, ...]
    [ WHERE condition ]
    [ GROUP BY user_id_column[, expression] [, ...] ]
    [ HAVING condition ]

final_select:
    SELECT [expression, ...] | COUNT | COUNT_DISTINCT | SUM | AVG | STDDEV
    FROM table_reference [, ...]
    [ WHERE condition ]
    [ GROUP BY expression [, ...] ]
    [ HAVING COUNT | COUNT_DISTINCT | SUM | AVG | STDDEV | condition ]
    [ ORDER BY column_list ASC | DESC ]
    [ OFFSET literal ]
    [ LIMIT literal ]

expression:
    column_name [, ...] | expression AS alias | aggregation_functions |
    window_functions_on_user_id | scalar_function | CASE | column_name math_expression [,
    expression]

window_functions_on_user_id:
```

```
function () OVER (PARTITION BY user_id_column, [column_name] [ORDER BY column_list  
ASC|DESC])
```

Note

差分プライバシークエリの構造と構文については、次の点に注意してください。

- サブクエリはサポートされていません。
- テーブルや CTE に差分プライバシーで保護されたデータが含まれている場合は、テーブル共通式 (CTE) がユーザー識別子列を出力する必要があります。フィルター、グループ化、集計はユーザーレベルで行う必要があります。
- Final_select では、COUNT DISTINCT、COUNT、SUM、AVG、STDDEV 集約関数が使用できます。

差分プライバシーでサポートされる SQL キーワードの詳細については、「[AWS Clean Rooms 差分プライバシーの SQL 機能](#)」を参照してください。

ID マッピングテーブル分析ルール

では AWS Clean Rooms、ID マッピングテーブル分析ルールはスタンドアロン分析ルールではありません。このタイプの分析ルールは、によって管理 AWS Clean Rooms され、クエリを容易にするために異なる ID データを結合するために使用されます。ID マッピングテーブルに自動的に追加され、編集することはできません。これは、それらの分析ルールが同種である限り、コラボレーション内の他の分析ルールの動作を継承します。

ID マッピングテーブル分析ルールにより、ID マッピングテーブルにセキュリティが適用されます。これにより、コラボレーションメンバーは、ID マッピングテーブルを使用して、2 人のメンバーのデータセット間の重複しない母集団を直接選択または検査することが制限されます。ID マッピングテーブル分析ルールは、他の分析ルールを使用したクエリで暗黙的に使用される場合に、ID マッピングテーブル内の機密データを保護するために使用されます。

ID マッピングテーブル分析ルールを使用すると、は拡張 SQL で ID マッピングテーブルの両側に重複 AWS Clean Rooms を適用します。これにより、次のタスクを実行できます。

- JOIN ステートメントで ID マッピングテーブルの重複を使用します。

AWS Clean Rooms は INNER、ID マッピングテーブルが重複している場合、ID マッピングテーブルで LEFT、または RIGHT 結合を許可します。機密マッピング情報を保護するには、ID マッ

ピングテーブルが常にJOINオペレーションのinner「」側にある必要があります。たとえば、次のJOINオペレーションは有効です。

- table LEFT JOIN id_mapping_table
- id_mapping_table RIGHT JOIN table
- table INNER JOIN id_mapping_table

以下のJOINオペレーションは無効です。

- id_mapping_table LEFT JOIN table
- table RIGHT JOIN id_mapping_table

これにより、データセット内で対応する一致がないマッピングレコードが公開されるのを防ぐことができます。このような操作を許可すると、他のコラボレーションメンバーのデータマッピングに関する機密情報が公開される可能性があります。

- JOIN ステートメントでマッピングテーブル列を使用します。

マッピングテーブル列は、SELECT、WHERE、HAVING、GROUP BY、または ORDER BY ステートメントでは使用できません (ソース ID 名前空間の関連付けまたはターゲット ID 名前空間の関連付けにより保護が変更される場合を除く)。

- 拡張 SQL では、は OUTER JOIN、暗黙的な JOIN、および CROSS AWS Clean Rooms もサポートしていますJOIN。これらの結合で重複要件を満たすことはできません。代わりに、AWS Clean Rooms を使用して、結合する必要がある列requireOverlapを指定します。

サポートされているクエリ構造と構文は、「[ID マッピングテーブルのクエリ構造と構文](#)」で定義されています。

「[ID マッピングテーブル分析ルールのクエリコントロール](#)」で定義されている分析ルールのパラメータには、クエリコントロールとクエリ結果コントロールがあります。そのクエリコントロールには、JOIN ステートメント (つまり requireOverlap) 内の ID マッピングテーブルの重複を要求する機能が含まれています。

トピック

- [ID マッピングテーブルのクエリ構造と構文](#)
- [ID マッピングテーブル分析ルールのクエリコントロール](#)
- [ID マッピングテーブル分析ルールの事前定義された構造](#)
- [ID マッピングテーブル分析ルール – 例](#)

ID マッピングテーブルのクエリ構造と構文

ID マッピングテーブル分析ルールがあるテーブルに対するクエリは、次の構文に従う必要があります。

```
--select_list_expression
SELECT
provider.data_col, consumer.data_col

--table_expression
FROM provider

JOIN idMappingTable idmt ON provider.id = idmt.sourceId

JOIN consumer ON consumer.id = idmt.targetId
```

コラボレーションテーブル

次の表は、AWS Clean Rooms コラボレーションに存在する設定済みテーブルを示しています。cr_drivers_license テーブルと cr_insurance テーブルの両方の [id] 列は、ID マッピングテーブルと一致する列を表します。

cr_drivers_license

id	driver_name	state_of_registration
1	Eduard	TX
2	Dana	MA
3	Gweneth	IL

cr_insurance

id	policyholder_email	policy_number

a	eduardo@internal.company.com	17f9d04e-f5be-4426-bdc4-250ed59c6529
b	gwen@internal.company.com	3f0092db-2316-48a8-8d44-09cf8f6e6c64
c	rosa@internal.company.com	d7692e84-3d3c-47b8-b46d-a0d5345f0601

ID マッピングテーブル

次の表は、cr_drivers_license テーブルと cr_insurance テーブルで一致する既存の ID マッピングテーブルを表しています。すべてのエントリに両方のコラボレーションテーブルの ID があるわけではありません。

cr_drivers_license_id	cr_insurance_id
1	a
2	null
3	b
null	c

ID マッピングテーブル分析ルールでは、重複するデータのセットに対してのみクエリの実行が許可されます。これは次のようになります。

cr_driver s_license_id	cr_insura nce_id	driver_name	state_of_ registration	policyhol der_email	policy_nu mber
1	a	Eduard	TX	eduardo@i nternal.c ompany.com	17f9d04e- f5be-4426 -bdc4-250 ed59c6529

3	b	Gweneth	IL	gwen@inte rnal.comp any.com	3f0092db- 2316-48a8 -8d44-09c f8f6e6c64

クエリの例

次の例は、ID マッピングテーブル結合の有効な場所を示しています。

```
-- Single ID mapping table
SELECT
  [ select_items ]FROM
    cr_drivers_license cr_dl
  [ INNER | LEFT ] JOIN cr_identity_mapping_table idmt ON idmt.cr_drivers_license_id
= cr_dl.id
  [ INNER | RIGHT ] JOIN cr_insurance cr_in          ON idmt.cr_insurance_id
= cr_in.id
;

-- Single ID mapping table (Subquery)
SELECT
  [ select_items ]FROM (
    SELECT
      [ select_items ]
    FROM
      cr_drivers_license cr_dl
      [ INNER | LEFT ] JOIN cr_identity_mapping_table idmt ON
idmt.cr_drivers_license_id = cr_dl.id
      [ INNER | RIGHT ] JOIN cr_insurance cr_in          ON idmt.cr_insurance_id
= cr_in.id
  )
;

-- Single ID mapping table (CTE)
WITH
  matched_ids AS (
    SELECT
      [ select_items ]
    FROM
      cr_drivers_license cr_dl
      [ INNER | LEFT ] JOIN cr_identity_mapping_table idmt ON
idmt.cr_drivers_license_id = cr_dl.id
      [ INNER | RIGHT ] JOIN cr_insurance cr_in          ON
idmt.cr_insurance_id      = cr_in.id
```

```
)SELECT  
[ select_items ]FROM  
matched_ids  
;
```

考慮事項

ID マッピングテーブルクエリの構造と構文については、次の点に注意してください。

- これを編集することはできません。
- デフォルトでは、ID マッピングテーブルに適用されます。
- コラボレーション内でソース ID とターゲット ID 名前空間の関連付けが使用されます。
- ID マッピングテーブルは、ID 名前空間から取得される列に対してデフォルトの保護を提供するようにデフォルトで設定されています。この設定を変更して、ID 名前空間 (sourceID または targetID) から取得される列をクエリ内の任意の場所で許可できます。詳細については、「[の ID 名前空間 AWS Clean Rooms](#)」を参照してください。
- ID マッピングテーブル分析ルールは、コラボレーション内の他の分析ルールの SQL 制限を継承します。

ID マッピングテーブル分析ルールのクエリコントロール

ID マッピングテーブルクエリコントロールを使用すると、はテーブル内の列を使用してテーブルをクエリする方法 AWS Clean Rooms を制御します。例えば、結合に使用する列と重複が必要な列を制御します。ID マッピングテーブル分析ルールには、JOIN を必要とせずに sourceID、targetID、またはその両方を射影できるようにする機能も含まれています。

以下の表では、それぞれのコントロールについて説明します。

コントロール	定義	使用方法
joinColumns	クエリを実行できるメンバーに INNER JOIN ステートメントでの使用を許可する列。	joinColumns は INNER JOIN 以外のクエリのどの部分でも使用できません。 詳細については、「 結合コントロール 」を参照してください。

コントロール	定義	使用方法
<code>dimensionColumns</code>	クエリを実行できるメンバーが SELECT ステートメントと GROUP BY ステートメントで利用できる列 (ある場合)。	<code>dimensionColumn</code> は、SELECT および GROUP BY で使用できます。 <code>dimensionColumn</code> は <code>joinKeys</code> として表示されることもあります。 <code>dimensionColumns</code> は、JOIN 句で角かっこで囲んで指定する場合にのみ使用できます。
<code>queryConstraints:RequireOverlap</code>	クエリを実行するために結合する必要がある ID マッピングテーブルの列。	これらの列は、ID マッピングテーブルとコラボレーションテーブルを結合するために使用する必要があります。

ID マッピングテーブル分析ルールの事前定義された構造

ID マッピングテーブル分析ルールの事前定義された構造には、`sourceID` と `targetID` に適用されるデフォルトの保護が含まれています。つまり、保護が適用された列をクエリで使用する必要があります。

ID マッピングテーブル分析ルールは、次の方法で設定できます。

- `sourceID` と `targetID` の両方が保護される

この設定では、`sourceID` と `targetID` はいずれも射影することはできません。ID マッピングテーブルが参照されている場合、`sourceID` と `targetID` は JOIN で使用する必要があります。

- `targetID` のみが保護される

この設定では、`targetID` を射影することはできません。ID マッピングテーブルが参照されている場合、`targetID` は JOIN で使用する必要があります。`sourceID` はクエリで使用できます。

- `sourceID` のみが保護される

この設定では、sourceID を射影することはできません。ID マッピングテーブルが参照されている場合、sourceID は JOIN で使用する必要があります。targetID はクエリで使用できます。

- sourceID と targetID のどちらも保護されない

この設定において、ID マッピングテーブルは、クエリで利用できる特定の適用の対象にはなりません。

次の例は、sourceID と targetID にデフォルトの保護が適用された ID マッピングテーブル分析ルール の事前定義された構造を示しています。この例では、ID マッピングテーブル分析ルールにより、sourceID 列と targetID 列の両方で INNER JOIN のみが許可されます。

```
{
  "joinColumns": [
    "source_id",
    "target_id"
  ],
  "queryConstraints": [
    {
      "requireOverlap": {
        "columns": [
          "source_id",
          "target_id"
        ]
      }
    }
  ],
  "dimensionColumns": [] // columns that can be used in SELECT and JOIN
}
```

次の例は、targetID に保護が適用された ID マッピングテーブル分析ルール の事前定義された構造を示しています。この例では、ID マッピングテーブル分析ルールにより、sourceID 列の INNER JOIN のみが許可されます。

```
{
  "joinColumns": [
    "source_id",
    "target_id"
  ],
  "queryConstraints": [
    {
```

```

    "requireOverlap": {
      "columns": [
        "target_id"
      ]
    }
  ],
  "dimensionColumns": [
    "source_id"
  ]
}

```

次の例は、sourceID に保護が適用された ID マッピングテーブル分析ルール of 事前定義された構造を示しています。この例では、ID マッピングテーブル分析ルールにより、targetID 列の INNER JOIN のみが許可されます。

```

{
  "joinColumns": [
    "source_id",
    "target_id"
  ],
  "queryConstraints": [
    {
      "requireOverlap": {
        "columns": [
          "source_id"
        ]
      }
    }
  ],
  "dimensionColumns": [
    "target_id"
  ]
}

```

次の例は、sourceID または targetID に保護が適用されない ID マッピングテーブル分析ルール of 事前定義された構造を示しています。この例では、ID マッピングテーブル分析ルールにより、sourceID 列と targetID 列の両方で INNER JOIN が許可されます。

```

{
  "joinColumns": [
    "source_id",

```

```
    "target_id"
  ],
  "queryConstraints": [
    {
      "requireOverlap": {
        "columns": []
      }
    }
  ],
  "dimensionColumns": [
    "source_id",
    "target_id"
  ]
}
```

ID マッピングテーブル分析ルール – 例

例えば、企業は個人を特定できる情報 (PII) を参照する長いウォーターフォールステートメントを記述するのではなく、ID マッピングテーブル分析ルールを使ってマルチパーティー LiveRamp トランスコーディングを使用できます。次の例は、ID マッピングテーブル分析ルール AWS Clean Rooms を使用して コラボレーションする方法を示しています。

A 社は、顧客と販売のデータを持つ広告主であり、これらのデータはソースとして使用されます。また、A 社はコラボレーションの当事者に代わってトランスコーディングを実行し、LiveRamp 認証情報を提供します。

B 社はイベントデータを持つパブリッシャーであり、このデータはターゲットとして使用されます。

Note

A 社または B 社は、LiveRamp のトランスコーディング認証情報を提供し、トランスコーディングを実行できます。

コラボレーションで ID マッピングテーブル分析を可能にするコラボレーションを作成するためには、各社は次のことを行います。

1. A 社がコラボレーションを作成し、メンバーシップを作成します。B 社を追加します。B 社もコラボレーションでメンバーシップを作成します。
2. A 社は、既存の ID 名前空間ソースを関連付けるか、AWS Clean Rooms コンソール AWS Entity Resolution を使用して 新しい ID 名前空間ソースを作成します。

A 社は、販売データと、ID マッピングテーブルの `sourceId` にキーが付けられた列を含む設定済みテーブルを作成します。

ID 名前空間ソースが、トランスコードするデータを提供します。

3. B 社は、既存の ID 名前空間ターゲットを関連付けるか、AWS Clean Rooms コンソール AWS Entity Resolution を使用して新しい ID 名前空間ターゲットを作成します。

B 社は、イベントデータと、ID マッピングテーブルの `targetId` にキーが付けられた列を含む設定済みテーブルを作成します。

ID 名前空間ターゲットはトランスコードするデータを提供せず、LiveRamp 設定に関連するメタデータのみを提供します。

4. A 社は、コラボレーションに関連付けられた 2 つの ID 名前空間を検出し、ID マッピングテーブルを作成してデータを入力します。
5. A 社は、ID マッピングテーブル上で結合することで、2 つのデータセット間でクエリを実行します。

```
--- this would be valid for Custom or List
SELECT provider.data_col, consumer.data_col
FROM provider
  JOIN idMappingTable-123123123123-myMappingWFName idmt
    ON provider.id = idmt.sourceId
  JOIN consumer
    ON consumer.id = idmt.targetId
```

AWS Clean Rooms 差分プライバシー

AWS Clean Rooms 差分プライバシーは、数回のクリックで直感的なコントロールで実装された数学に基づく手法により、ユーザーのプライバシーを保護するのに役立ちます。フルマネージド機能として、ユーザーの再識別を防ぐために、事前の差分プライバシーエクスペリエンスは必要ありません。は、個々のレベルのデータを保護するために、実行時に結果をクエリするために慎重に調整された量のノイズ AWS Clean Rooms を自動的に追加します。

AWS Clean Rooms 差分プライバシーは、幅広い分析クエリをサポートし、クエリ結果のわずかなエラーによって分析の有用性が損なわれないさまざまなユースケースに適しています。これにより、パートナーは広告キャンペーン、投資決定、臨床研究などについて、ビジネスに不可欠なインサイトを生成できます。しかも、パートナーによる追加設定は必要ありません。

AWS Clean Rooms 差分プライバシーは、スカラー関数や数学演算子記号を悪意のある方法で使用するオーバーフローや無効なキャストエラーから保護します。

AWS Clean Rooms 差分プライバシーの詳細については、以下のトピックを参照してください。

トピック

- [差分プライバシー](#)
- [での差分プライバシーの AWS Clean Rooms 仕組み](#)
- [差分プライバシーポリシー](#)
- [AWS Clean Rooms 差分プライバシーの SQL 機能](#)
- [差分プライバシークエリのヒントと例](#)
- [AWS Clean Rooms 差分プライバシーの制限](#)

差分プライバシー

差分プライバシーでは、集約されたインサイトのみが許可され、それらのインサイトにおける個人のデータの関与をわかりにくくします。差分プライバシーは、特定の個人について学習した結果を受け取ることができるメンバーからコラボレーションデータを保護します。差分プライバシーがなければ、結果を受け取ることができるメンバーは、個人に関するレコードを追加または削除したり、クエリ結果の違いを確認したりして、個々のユーザーデータを推測しようとする可能性があります。

差分プライバシーをオンにすると、特定の量のノイズがクエリ結果に追加され、個々のユーザーの関与がわかりにくくなります。データセットから個人に関するレコードを削除した後に、結果を受け取ることができるメンバーがクエリ結果の違いを観察しようとする、クエリ結果の変動性は、個人のデータの識別を防ぐのに役立ちます。AWS Clean Rooms 差分プライバシーは、によって開発された実証済みの正しいサンプラー実装である [SampCert](#) サンプラーを使用します AWS。

での差分プライバシーの AWS Clean Rooms 仕組み

で差分プライバシーを有効にするワークフローでは、[ワークフローを完了する AWS Clean Rooms](#) ときに以下の追加ステップ AWS Clean Rooms が必要です。

1. [カスタム分析ルールを追加する](#) ときに、差分プライバシーを有効にします。
2. [コラボレーションの差分プライバシーポリシーを設定して](#)、差分プライバシーで保護されているデータテーブルをクエリに使用できるようにします。

これらのステップを完了すると、クエリを実行できるメンバーは、差分プライバシー保護データに対するクエリの実行を開始できます。差分プライバシーポリシーに準拠した結果を AWS Clean Rooms 返します。AWS Clean Rooms 差分プライバシーは、車の現在の燃料レベルを示す車のガスゲージと同様に、実行できる残りのクエリの推定数を追跡します。クエリを実行できるメンバーが実行できるクエリ数は、[差分プライバシーポリシー](#) で設定されている [プライバシー予算] と [クエリごとに追加されるノイズ] パラメータによって制限されます。

考慮事項

差分プライバシーを使用する場合は AWS Clean Rooms、次の点を考慮してください。

- 結果を受け取ることができるメンバーは、差分プライバシーを使用できません。これらのメンバーは、設定したテーブルの差分プライバシーをオフにしたカスタム分析ルールを設定します。
- クエリを実行できるメンバーは、2 つ以上のデータプロバイダーの差分プライバシーがオンになっている場合、テーブルを結合できません。

差分プライバシーポリシー

差分プライバシーポリシーは、クエリを実行できるメンバーがコラボレーション内で実行できる集計関数の数を制御します。プライバシー予算は、コラボレーションのすべてのテーブルに適用される共通の有限リソースを定義します。クエリごとに追加されるノイズによって、プライバシー予算が枯渇する速度が決まります。

差分プライバシーで保護されたテーブルをクエリに使用できるようにするには、差分プライバシーポリシーが必要です。これはコラボレーションの 1 回限りのステップで、次の 2 つの入力が含まれます。

- プライバシー予算 – プライバシー予算はイプシロン単位で定量化され、プライバシー保護のレベルを制御します。これは、コラボレーションにおいて差分プライバシーで保護されるすべてのテーブルに適用される共通かつ有限のリソースです。その目的は、情報が複数のテーブルに存在する可能性があるユーザーのプライバシーを保護することです。

プライバシー予算は、テーブルに対してクエリが実行されるたびに消費されます。プライバシー予算が使い果たされると、クエリを実行できるコラボレーションメンバーは、プライバシー予算が増えるか更新されるまで追加のクエリを実行できなくなります。プライバシー予算を大きく設定することで、結果を受け取ることができるメンバーは、データ内の個人に関する不確実性を減らすことができます。ビジネス上の意思決定者と相談した上で、コラボレーション要件とプライバシーニーズのバランスを考慮して、プライバシー予算を選択してください。

コラボレーションに定期的に新しいデータを取り込む予定がある場合は、[プライバシー予算を毎月更新] を選択すると、1 か月ごとに新しいプライバシー予算が自動的に作成されます。このオプションを選択すると、更新を繰り返してクエリを繰り返し実行したときに、データ行に関する任意の量の情報を公開できます。プライバシー予算が更新されるたびに同じ行が繰り返しクエリされる場合は、このオプションを選択しないでください。

- クエリごとに追加されるノイズは、関与をわかりにくくするユーザーの数で測定されます。この値によって、プライバシー予算がどの程度枯渇するかが決まります。ノイズ値を大きくすると、プライバシー予算が枯渇する速度が下がるため、データに対して実行できるクエリの数が増えます。ただし、これは正確性の低いデータインサイトの公開とのバランスを取る必要があります。この値を設定するときは、コラボレーションに関するインサイトに必要な精度を考慮してください。

デフォルトの差分プライバシーポリシーを使用すると、ユースケースに応じて設定をすばやく完了したり、差分プライバシーポリシーをカスタマイズしたりできます。AWS Clean Rooms 差分プライバシーは、ポリシーを設定する直感的なコントロールを提供します。AWS Clean Rooms 差分プライバシーを使用すると、データに対するすべてのクエリで可能な集約数の観点からユーティリティをプレビューし、データコラボレーションで実行できるクエリの数进行推定できます。

インタラクティブな例を見れば、[プライバシー予算] と [クエリごとに追加されるノイズ] の値が異なると、さまざまな種類の SQL クエリの結果にどのような影響があるかを理解できます。一般的には、プライバシーに関するニーズと、許可するクエリの数やそれらのクエリの精度とのバランスを取る必要があります。[プライバシー予算] を小さくしたり、[クエリごとに追加されるノイズ] を大きくしたりすると、ユーザーのプライバシー保護は強化されますが、コラボレーションパートナーにとって意味のあるインサイトは得られません。

[クエリごとに追加されるノイズ] のパラメータをそのままにして、[プライバシー予算] を増やすと、クエリを実行できるメンバーは、コラボレーション内のテーブルに対してより多くの集計を実行できます。コラボレーション中はいつでも [プライバシー予算] を増やすことができます。[クエリごとに追加されるノイズ] パラメータをそのままにして、[プライバシー予算] を減らすと、クエリを実行できるメンバーが実行できる集計の数が減ります。クエリを実行できるメンバーがデータの分析を開始した後は、[プライバシー予算] を減らすことはできません。

[プライバシー予算] の入力をそのままにして、[クエリごとに追加されるノイズ] を増やすと、クエリを実行できるメンバーは、コラボレーション内のテーブルに対してより多くの集計を実行できます。[プライバシー予算] の入力をそのままにして、[クエリごとに追加されるノイズ] を減らすと、クエリを実行できるメンバーが実行できる集計の数が減ります。[クエリごとに追加されるノイズ] は、コラボレーション中いつでも増減できます。

差分プライバシーポリシーは、プライバシー予算テンプレート API アクションによって管理されます。

AWS Clean Rooms 差分プライバシーの SQL 機能

AWS Clean Rooms 差分プライバシーは、汎用クエリ構造を使用して複雑な SQL クエリをサポートします。カスタム分析テンプレートは、差分プライバシーで保護されたテーブルで実行できるよう、この構造に対して検証されます。次の表は、どの関数がサポートされているかを示しています。詳細については「[クエリの構造と構文](#)」を参照してください。

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

SAMPLEVALUE

Supported

functions

to:

Approximation

APPROXIMA

COUNT,

PERCENTIL

DISTINCT,

STDDEV,

and

privacy

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
pruned
tables
most
result
in
data
with
user
level
| 関
数

Category
終
概ELECT
勤
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
re MAX.
Yo 関
sh 数
• write
• MEDIAN
the 関
SELECT
数
expressio
• MIN
n 関
in 数
those
CTEs
using

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
・`PERCENTIL
ile_Quant
ile`関
数..
format
STDDEV_SA
MP
お
よ
び
STDDEV_PO
P

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

関

数

• SUM

お

よ

び

SUM

DISTINCT

関

数

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• VAR_SAMP

お

よ

び

VAR_POP

関

数

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

Supported

With,

WITH

condition

that query

CTEs

using

different

ial

privacy

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

protected

tables

must

result

in

data

with

user-

leve

|

records.

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

You

should

write

the

SELECT

expressio

n

in

those

CTEs

using

Category

終

極ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

`SELECT

userIdent

ifierColu

mn...'

format.

Category	
終極	
概観	ELECT
基	
通	
式	
(CTE)	
で	
サ	
ポ	
ー	
ト	
さ	
れ	
て	
い	
る	
SQL	
コ	
ン	
ス	
ト	
ラ	
ク	
ト	
Subquery	SELECT
can	HAVING
have	
• JOIN	
any	
• JOIN	
subquery	
条件	
that	
doesn't	
• FROM	
reference	
WHERE	
different	
ial	

Category
SELECT
JOIN
通式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

privacy
relations
in
these
construct
s.
You
can
have
any
subquery

Category
終
極
基
礎
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

that
reference
s
different
ial
privacy
relations
in
a
FROM
and

Category
終
極
基
礎
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

JOIN
clause
only.

Category
終
極ELECT
勤
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

INNER
JOIN
識別
LEFT
JOIN
子
列
の
等
価
結
果
ア

Category
終
極ELECT
勤
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
合
ン
で
チ
あ
結
る
合
• JOIN
RIGHT
関
JOIN
数
• FULL
の
JOIN
み
• [JOIN]
が
OR
サ

Category
終
極ELECT
勤
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ポ
演
ト算
さ子
れCROSS
差JOIN
分
プ
ラ
イ
バ
シー

Category
終
極ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
が
有
効
に
な
っ
て
い
る
複
数
の

Category
を
終
極
ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
テ
ー
ブ
ル
を
ク
エ
リ
す
る
場
合

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
は
必
須
で
あ
る
と
い
う
条
件

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
で
サ
ポ
ー
ト
さ
れ
ま
す。
必
須
の

Category
終
極ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
等
価
結
合
条
件
が
正
し
い
こ

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
と
を
確
認
し
て
く
だ
さ
い。
テー

Category
SELECT
データベース
通式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ブ
ル
所
有
者
が
す
べ
て
の
テー

Category
最終
概ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ブ
ル
に
同
じ
ユー
ザー
ID
列
を
設

Category
SELECT
データベース
通式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
定
し
て、
ユー
ザー
の
定
義
が
テー
ブ

Category
最終
概ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ル
間
で
一
貫
し
て
い
る
こ
と

Category

終

極ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

を

確

認

し

ま

す。

差

分

プ

ラ

Category
を
ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
イ
バ
シー
を
有
効
に
し
て
2
つ

Category
最終
概
ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
以
上
の
リ
レー
シ
ョ
ン
を
組
み
合

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

わ

せ

る

場

合、CROSS

JOIN

関

数

は

サ

ポー

Category
で
ELECT
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ト
さ
れ
ま
せ
ん。

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

UNION,

UNION

ALL,

INTERSECT

”

EXCEPT

||

MINUS

(these

are

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

synonyms))

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

Window

Supported

Supported

数with

the

• AVG condition

that

the

user

identify

Category
最終
概
ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
r 関
数
column
in COUNT
the
ウィ
window
function'
s ウ
partition
関
clause
数
is

Category
終
極ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
requirement_DIST
when
your
query
a ウ
relation
with
different
DENSE_RANK
ial
privacy
ン

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

turned

on

ウ

関

数

• FIRST_VAL

UE

ウィ

ン

ド

ウ

Category

SELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

関

数

- LAG

ウィ

ン

ド

ウ

関

数

Category

SELECT

関数
通式
(CTE)

で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

- LAST_VALUE
ウィ
ン
ド
関
数
- LEAD
ウィ
ン

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

ド

ウ

関

数

数

• MAX

ウィ

ン

ド

ウ

関

数

Category

SELECT

関数
通式
(CTE)

で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

- MEDIAN
ウィ
ン
ド
関
数
- MIN
ウィ
ン
ド

Category

SELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

ウ

関

数

- NTH_VALUE

ウィ

ン

ド

ウ

関

数

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• STDDEV_SA

MP

お

よ

び

STDDEV_PO

P

ウィ

ン

ド

ウ

Category
関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

関
数
(STDDEV_S
AMP
お
よ
び
STDDEV
は
シ
ノ

Category
終
極ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ニ
ム
で
す)
• SUM
ウィ
ン
ド
ウ
関
数

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• VAR_SAMP

お

よ

び

VAR_POP

ウィ

ン

ド

ウ

関

数

Category
SELECT
WITH
TABLE
FUNCTION
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

(VAR_SAMP
お
よ
び
VARIANCE
は
シ
ノ
ニ
ム

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

で

す)

ラ

ン

ク

付

け

関

数

Category

終
極
ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- DENSE_RANK

ウィ

ン

ド

ウ

関

数

- NTILE

ウィ

ン

Category

SELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

ド

ウ

関

数

- PERCENT_R

ANK

ウィ

ン

ド

ウ

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

関

数

• RANK

ウィ

ン

ド

ウ

関

数

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• ROW_NUMBE

R

ウィ

ン

ド

ウ

関

数

Category

終極
ELECT
式
通式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

ANSI

are
supported
ns式

- COALESCE
式
- GREATEST
お
よ
び

Category

SELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

LEAST

関

数

• NVL

お

よ

び

COALESCE

関

数

Category
関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

• NVL2
関
数

• NULLIF
関
数

Category

終

ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

Condition

and

supported

can't

be

used

because

they

require

subquerie

Category

終

極ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

s.ン

AIRツ

others

are

are

supported

. 条

件

• BETWEEN

範

囲

Category
終
極
基
礎
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
条
件
• Null
条
件

Category

終

極ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

Alt-

are

supported

ザ

ク

シヨ

ン

に

お

け

Category
最終
概ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
る
日
付
お
よ
び
時
刻
関
数

Category

終

概ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- 連
結
演
算
子

- ADD_MONTH
S
関
数

Category

関

関
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- CONVERT_T
IMEZONE

関

数

- CURRENT_D
ATE

関

数

Category

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

関数

• DATEADD

関

数

• DATEDIFF

関

数

• DATE_PART

関

数

Category

関数
ELECT

関
通
式
(CTE)

で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

- DATE_TRUNC
関
数
- EXTRACT
関
数
- TO_TIMESTAMP

Category
関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
関
数
・ 日
付
関
数
ま
た
は
タ
イ

Category
最終
概ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
ム
ス
タ
ン
プ
関
数
の
日
付

Category
SELECT
WITH
CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
部
分

Category

終

極ELECT

勤

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

SQLing

functions

supported

演

算

子

• BTRIM

関

数

Category

SELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- CHAR_LENGTH
関
数
- CHARACTER_LENGTH
関
数

Category

関数
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• CONCAT

関

数

• LEFT

関

数

お

よ

び

RIGHT

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

関

数

- LEN

関

数

- LENGTH

関

数

Category

関数
ELECT

関数
通式
(CTE)

で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

- LOWER
関数
- LPAD
関数
および
RPAD

Category

関

関

関

関

関

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

関

数

- LTRIM

関

数

- POSITION

関

数

- REGEXP_CO

UNT

Category

関

関

関

関

関

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

関

数

- REGEXP_INSTR

関

数

- REGEXP_REPLACE

関

数

Category

関数
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• REGEXP_SU

BSTR

関

数

• REPEAT

関

数

• REPLACE

関

数

Category

関数
ELECT

関
通
式
(CTE)

で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

- REVERSE
関
数
- RTRIM
関
数
- SPLIT_PAR
T
関
数

Category

関

数ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- SUBSTRING

関

数

- TRANSLATE

関

数

- TRIM

関

数

Category
関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
・ UPPER
関
数

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

CAST

関

supported

9 TO_CHAR

functions

• TO_DATE

関

数

• TO_NUMBER

• 日

時

Category
SELECT
WITH
通式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
形
式
の
文
字
列
・数
値
形
式
の

Category
最終
概ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト
文
字
列

Category	
関数	
基本関数	
SELECT	
関数	
通式	
式	
(CTE)	
で	
サ	
ポ	
ー	
ト	
さ	
れ	
て	
い	
る	
SQL	
コ	
ン	
ス	
ト	
ラ	
ク	
ト	
関数	
AWES_ENCRY	
関数	
関数	
supported	
AWES_DECRY	
PT	
• ENCODE	
• DECODE	
• MD5	
関数	
関数	

Category
関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

- SHA1
関
数
- SHA2
関
数
- XX_HASH64

Category
終極
ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

Mathematical
symbols
and
@

Category

関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

MATHS
functions
supported

- ACOS
関
数
- ASIN
関
数

Category

関数

ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• ATAN

関

数

• ATAN2

関

数

• CBRT

関

数

Category

関数
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• CEILING

(ま

た

は

CEIL)

関

数

• COS

関

数

Category

関数

ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• COT

関

数

• DEGREES

関

数

• LTRIM

関

数

Category

関数
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- EXP
関
数
- FLOOR
関
数
- LN
関
数

Category

関数

ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• LOG

関

数

• MOD

関

数

• PI

関

数

Category

関数
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• POWER

関

数

• RADIANS

関

数

• RANDOM

関

数

Category

関数
ELECT

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

• ROUND

関

数

• SIGN

関

数

• SIN

関

数

Category
関数
ELECT
関
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

• SQRT
関
数

• TRUNC
関
数

Category

関数

概観

関

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

MINBYEX、

functions、UNBASE64

supported

• HEX

• HLL_SKETC
H_AGG、

• HLL_SKETC
H_ESTIMAT
E

• HLL_UNION

Category
終
極
基
礎
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

• HLL_UNION
_AGG

ADD_JSON
are GET_JSON_
supported
OBJECT

Category

最終更新
2023年10月
ELECT
基
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

Not supported
ARRAY_CON
TAINS

- ARRAY_DISTINCT
- ARRAY_EXCEPT
- ARRAY_INTERSECT
- ARRAY_JOIN

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

- ARRAY_REMOV
- ARRAY_SORT
- ARRAY_UNION

Category
SELECT
JOIN
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

NO JOIN
GROUPED
ROLLUP,
CUBE

Category

終

概ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

Supported

Operation

the

condition

that

an

ORDER

BY

clause

is

Category

終

極ELECT

基

通

式

(CTE)

で

サ

ポ

ー

ト

さ

れ

て

い

る

SQL

コ

ン

ス

ト

ラ

ク

ト

only

supported

in

a

window

function'

s

partition

clause

when

querying

Category
SELECT
with
different
privacy
turned
on.

Category
SELECT
JOIN
通
式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

ROW,
OFFSET
supported
CTEs
using
different
ial
privacy
protected
tables

Category
SELECT
WITH
TABLE
FUNCTIONS
(CTE)
with
support
for
SQL
columns
and
aliasing
Supported
functions
on
aggregate
functions

Category
SELECT
関数
通式
(CTE)
で
サ
ポ
ー
ト
さ
れ
て
い
る
SQL
コ
ン
ス
ト
ラ
ク
ト

Supported
functions
within
aggregate
functions

サポートされていない SQL コンストラクトの一般的な代替方法

Category	SQL コンストラクト	代替
Window 関数	• LISTAGG • PERCENTILE_CONT • PERCENTILE_DISC	You can use the equivalent aggregate function with GROUP BY.
Mathematical operator symbols	• \$column / 2 • \$column / 2 • \$column ^ 2	• CBRT • SQRT • POWER(\$column, 2)
Scalar functions	• SYSDATE • \$column::integer • convert(type, \$column)	• CURRENT_DATE • CAST \$column AS integer • CAST \$column AS type
Literals	INTERVAL '1 SECOND'	INTERVAL '1' SECOND
Row limiting	TOP n	LIMIT n
Join	• USING • NATURAL	ON clause should explicitly contain a join criterion.

差分プライバシークエリのヒントと例

AWS Clean Rooms 差分プライバシーは、[汎用クエリ構造](#)を使用して、データ準備のための共通テーブル式 (CTEs) や、`、`などの一般的に使用される集計関数などCOUNT、さまざまな SQL コンストラクトをサポートしますSUM。実行時にクエリ結果を集計するためのノイズを追加して、データ内の可能なユーザーの寄与を難読化するために、AWS Clean Rooms 差分プライバシーでは、最後の集計関数SELECT statementをユーザーレベルのデータで実行する必要があります。

次の例では、athletic_brand_sales データを持つスポーツブランドとのコラボレーションで差分プライバシーを使用してデータを保護したいと考えているメディアパブリッシャーのsocialco_impressions と socialco_users という名前の 2 つのテーブルを使用しています。メディアパブリッシャーは、AWS Clean Roomsで差分プライバシーを有効にして、user_id 列をユーザー識別子列として設定しています。広告主は差分プライバシー保護を必要としないため、組み合わせたデータに対して CTE を使用してクエリを実行したいと考えています。CTE では差分プライ

バシーで保護されたテーブルを使用しているため、広告主は保護されているテーブルのユーザー ID 列を CTE 列のリストに含め、保護対象のテーブルをユーザー ID 列に結合します。

```
WITH matches_table AS(
    SELECT si.user_id, si.campaign_id, s.sale_id, s.sale_price
    FROM socialco_impressions si
    JOIN socialco_users su
        ON su.user_id = si.user_id
    JOIN athletic_brand_sales s
        ON s.emailsha256 = su.emailsha256
    WHERE s.timestamp > si.timestamp

    UNION ALL

    SELECT si.user_id, si.campaign_id, s.sale_id, s.sale_price
    FROM socialco_impressions si
    JOIN socialco_users su
        ON su.user_id = si.user_id
    JOIN athletic_brand_sales s
        ON s.phonesha256 = su.phonesha256
    WHERE s.timestamp > si.timestamp
)

SELECT COUNT (DISTINCT user_id) as unique_users
FROM matches_table
GROUP BY campaign_id
ORDER BY COUNT (DISTINCT user_id) DESC
LIMIT 5
```

同様に、差分プライバシーで保護されたデータテーブルでウィンドウ関数を実行する場合は、PARTITION BY 句にユーザー ID 列を含める必要があります。

```
ROW_NUMBER() OVER (PARTITION BY conversion_id, user_id ORDER BY match_type, match_age)
AS row
```

AWS Clean Rooms 差分プライバシーの制限

AWS Clean Rooms 差分プライバシーは、以下の状況には対応しません。

1. AWS Clean Rooms 差分プライバシーは、Amazon S3-backed AWS Glue テーブルを使用したクエリのみをサポートします。Snowflake または Amazon Athena テーブルを使用したクエリはサポートされていません。

2. AWS Clean Rooms 差分プライバシーはタイミング攻撃に対処しません。例えば、これらの攻撃は、個々のユーザーが大量の行を入力し、このユーザーを追加または削除するとクエリの計算時間が大幅に変化するようなシナリオで発生する可能性があります。
3. AWS Clean Rooms 差分プライバシーは、特定の SQL コンストラクトの使用が原因で実行時にオーバーフローまたは無効なキャストエラーが発生する可能性がある場合、差分プライバシーを保証しません。

次の表は、ランタイムエラーを発生させる可能性があり、分析テンプレートで検証する必要がある一部 (すべてではありません) の SQL コンストラクトのリストです。このようなランタイムエラーの可能性を最小限に抑える分析テンプレートを承認し、クエリログを定期的にレビューして、クエリがコラボレーション契約と整合しているかどうかを確認することをお勧めします。

次の SQL コンストラクトは、オーバーフローエラーに対して脆弱です。

Category	Spark SQL 分析エンジンのオーバーフローエラーに対して脆弱な SQL コンストラクト
Aggregate functions	• AVG • SUM/SUM_DISTINCT
Data type formatting functions	• TO_TIMESTAMP • TO_DATE
Date and time functions	• ADD_MONTHS • DATEADD • DATEDIFF
Math functions	• +, -, *, / • POWER
String functions	• • CONCAT • REPEAT
Window functions	• AVG • SUM

4. CAST データ形式設定関数は、無効なキャストエラーに対して脆弱です。

[CloudWatch を設定してロググループのメトリクスフィルターを作成](#)し、そのメトリクスフィルターで [CloudWatch アラームを作成](#)して、オーバーフローまたはキャストエラーが発生した可能性がある場合にアラートを受信できます。

具体的には、CastError、OverflowError、ConversionError の各エラーコードをモニタリングする必要があります。これらのエラーコードの存在は、サイドチャネル攻撃の可能性を示していますが、誤った SQL クエリを示している可能性もあります。

詳細については、「[分析ログイン AWS Clean Rooms](#)」を参照してください。

AWS クリーンルーム ML

AWS Clean Rooms ML では、2 人以上の関係者がデータで機械学習モデルを実行でき、相互にデータを共有する必要はありません。このサービスは、データ所有者がデータとモデル IP を保護できるようにするプライバシー強化コントロールを提供します。AWS オーサリングされたモデルを使用するか、独自のカスタムモデルを取り込むことができます。

この動作の詳細な説明については、「[クロスアカウントジョブ](#)」を参照してください。

Clean Rooms ML モデルの機能の詳細については、以下のトピックを参照してください。

トピック

- [AWS Clean Rooms ML の用語](#)
- [AWS Clean Rooms ML が AWS モデルと連携する方法](#)
- [AWS Clean Rooms ML がカスタムモデルと連携する方法](#)
- [AWS Clean Rooms ML のモデル](#)
- [Clean Rooms ML のカスタムモデル](#)

AWS Clean Rooms ML の用語

Clean Rooms ML を使用する場合は、次の用語を理解することが重要です。

- トレーニングデータプロバイダー – トレーニングデータを提供し、類似モデルを作成および設定し、その類似モデルをコラボレーションに関連付ける関係者。

- シードデータプロバイダー – シードデータを提供し、類似セグメントを生成し、その類似セグメントをエクスポートする関係者。
- トレーニングデータ – 類似モデルの生成に使用されるトレーニングデータプロバイダーのデータ。トレーニングデータは、ユーザーの行動の類似性を測定するために使用されます。

トレーニングデータには、ユーザー ID、項目 ID、タイムスタンプ列が含まれている必要があります。オプションで、トレーニングデータには数値特徴量またはカテゴリ別特徴量として他のインタラクションを含めることができます。インタラクションの例としては、視聴した動画のリスト、購入したアイテム、読んだ記事などがあります。

- シードデータ – 類似セグメントの作成に使用されるシードデータプロバイダーのデータ。シードデータは直接提供することも、AWS Clean Rooms クエリの結果から取得することもできます。類似セグメントの出力は、トレーニングデータに含まれるシードユーザーに最も近いユーザーの集合です。
- 類似モデル – 他のデータセット内の類似ユーザーを見つけるために使用されるトレーニングデータの機械学習モデル。

API を使用する場合、オーディエンスモデルという用語は類似モデルと同じ意味で使用されます。例えば、[CreateAudienceModel](#) API を使用して類似モデルを作成します。

- 類似セグメント – シードデータに最も近いトレーニングデータのサブセット。

API を使用するときは、[StartAudienceGenerationJob](#) API を使用して類似セグメントを作成します。

トレーニングデータプロバイダーのデータがシードデータプロバイダーと共有されることはなく、シードデータプロバイダーのデータがトレーニングデータプロバイダーと共有されることもありません。類似セグメントの出力はトレーニングデータプロバイダーと共有されますが、シードデータプロバイダーと共有されることはありません。

AWS Clean Rooms ML が AWS モデルと連携する方法

▼ How it works



Step 1. Import training data

Select an AWS Glue table and identify the columns to include in your lookalike model.

[Create training dataset](#)



Step 2. Create a lookalike model

Train a model from your datasets that advertisers will use to match to their users.

[Create lookalike model](#)



Step 3. Configure a lookalike model

Determine how the lookalike model is trained.

[Configure lookalike model](#)



Step 4. Associate a lookalike model

From the Collaborations page, choose which lookalike models to include in each collaboration.

[View collaborations](#)

類似モデルを使用するには、トレーニングデータプロバイダーとシードデータプロバイダーの 2 つの関係者が、データをコラボレーションに取り込む AWS Clean Rooms ために順番に作業する必要があります。トレーニングデータプロバイダーが最初に完了しなければならないワークフローは次のとおりです。

1. トレーニングデータプロバイダーのデータは、ユーザーとアイテムのインタラクション AWS Glue のデータカタログテーブルに保存する必要があります。少なくとも、トレーニングデータにはユーザー ID 列、インタラクション ID 列、およびタイムスタンプ列が含まれている必要があります。
2. トレーニングデータプロバイダーは、トレーニングデータを に登録します AWS Clean Rooms。
3. トレーニングデータプロバイダーは、複数のシードデータプロバイダーと共有できる類似モデルを作成します。類似モデルはディープニューラルネットワークであり、トレーニングに最大 24 時間かかることがあります。このモデルは自動的に再トレーニングされないため、毎週再トレーニングすることをお勧めします。
4. トレーニングデータプロバイダーは、関連メトリクスを共有するかどうかや、出力セグメントの Amazon S3 ロケーションなど、類似モデルの設定を行います。トレーニングデータプロバイダーは、1 つの類似モデルから複数の設定済み類似モデルを作成できます。
5. トレーニングデータプロバイダーは、設定したオーディエンスモデルをシードデータプロバイダーと共有されているコラボレーションに関連付けます。

シードデータプロバイダーが次に完了しなければならないワークフローは次のとおりです。

1. シードデータプロバイダーのデータは Amazon S3 バケットに保存することも、クエリの結果から取得することもできます。

2. シードデータプロバイダーは、トレーニングデータプロバイダーと共有するコラボレーションを開きます。
3. シードデータプロバイダーは、コラボレーションページの [Clean Rooms ML] タブから類似セグメントを作成します。
4. シードデータプロバイダーは、関連性メトリクスが共有されていれば、それを評価し、AWS Clean Roomsの外部で使用するために類似セグメントをエクスポートできます。

AWS Clean Rooms ML がカスタムモデルと連携する方法

Clean Rooms ML を使用すると、コラボレーションのメンバーは Amazon ECR に保存されているドッカー化されたカスタムモデルアルゴリズムを使用して、データを共同で分析できます。これを行うには、モデルプロバイダーがイメージを作成し、Amazon ECR に保存する必要があります。[Amazon Elastic Container Registry ユーザーガイド](#)の手順に従って、カスタム ML モデルを含むプライベートリポジトリを作成します。

コラボレーションのメンバーは、適切なアクセス許可があれば、モデルプロバイダーになることができます。コラボレーションのすべてのメンバーは、トレーニングデータ、推論データ、またはその両方をモデルに提供できます。このガイドでは、データを提供するメンバーをデータプロバイダーと呼びます。コラボレーションを作成するメンバーはコラボレーション作成者であり、このメンバーはモデルプロバイダー、データプロバイダーの 1 つ、またはその両方です。

最も高いレベルでは、カスタム ML モデリングを実行するために完了する必要がある手順は次のとおりです。

1. コラボレーション作成者はコラボレーションを作成し、各メンバーに適切なメンバー機能と支払い設定を割り当てます。コラボレーションクリエイターは、コラボレーションの作成後に更新できないため、モデル出力を受信するか、推論結果を受信する機能をこのステップの適切なメンバーに割り当てる必要があります。詳細については、「[AWS Clean Rooms ML でのコラボレーションの作成と参加](#)」を参照してください。
2. モデルプロバイダーは、コンテナ化された ML モデルを設定してコラボレーションに関連付け、エクスポートされたデータにプライバシー制約が設定されていることを確認します。詳細については、「[AWS Clean Rooms ML でのモデルアルゴリズムの設定](#)」を参照してください。
3. データプロバイダーは、コラボレーションにデータを提供し、プライバシーのニーズを確実に指定します。データプロバイダーは、モデルがデータにアクセスすることを許可する必要があります。詳細については、「[AWS Clean Rooms ML でのトレーニングデータの提供](#)」および「[AWS Clean Rooms ML で設定されたモデルアルゴリズムの関連付け](#)」を参照してください。

4. コラボレーションメンバーは、モデルアーティファクトまたは推論結果がエクスポートされる場所を定義する ML 設定を作成します。
5. コラボレーションメンバーは、トレーニングコンテナまたは推論コンテナに入力を提供する ML 入力チャネルを作成します。ML 入力チャネルは、モデルアルゴリズムのコンテキストで使用されるデータを定義するクエリです。
6. コラボレーションメンバーは、ML 入力チャネルと設定されたモデルアルゴリズムを使用してモデルトレーニングを呼び出します。詳細については、「[AWS Clean Rooms ML でのトレーニング済みモデルの作成](#)」を参照してください。
7. (オプション) モデルトレーナーはモデルエクスポートジョブを呼び出し、モデルアーティファクトがモデル結果レシーバーに送信されます。有効な ML 設定を持ち、モデル出力を受け取ることができるメンバーのみがモデルアーティファクトを受け取ることができます。詳細については、「[AWS Clean Rooms ML からのモデルアーティファクトのエクスポート](#)」を参照してください。
8. (オプション) コラボレーションメンバーは、ML 入力チャネル、トレーニング済みモデル ARN、推論設定済みモデルアルゴリズムを使用してモデル推論を呼び出します。推論結果は推論出力レシーバーに送信されます。有効な ML 設定を持ち、推論出力を受け取ることができるメンバーのみが推論結果を受け取ることができます。

モデルプロバイダーが完了する必要があるステップは次のとおりです。

1. SageMaker AI 互換 Amazon ECR Docker イメージを作成します。Clean Rooms ML は、SageMaker AI 互換のドッカーイメージのみをサポートしています。
2. SageMaker AI 互換の Docker イメージを作成したら、そのイメージを Amazon ECR にプッシュします。[Amazon Elastic Container Registry ユーザーガイド](#)の指示に従って、コンテナトレーニングイメージを作成します。
3. Clean Rooms ML で使用するモデルアルゴリズムを設定します。
 - a. Amazon ECR リポジトリリンクと、モデルアルゴリズムの設定に必要な引数を指定します。
 - b. Clean Rooms ML が Amazon ECR リポジトリにアクセスできるようにするサービスアクセスロールを提供します。
 - c. 設定されたモデルアルゴリズムをコラボレーションに関連付けます。これには、コンテナログ、障害ログ、CloudWatch メトリクスのコントロール、およびコンテナ結果からエクスポートできるデータ量に関する制限を定義するプライバシーポリシーの提供が含まれます。

カスタム ML モデルとコラボレーションするためにデータプロバイダーが完了する必要がある手順は次のとおりです。

1. カスタム分析ルールを使用して既存の AWS Glue テーブルを設定します。これにより、特定の一連の事前承認されたクエリまたは事前承認されたアカウントでデータを使用できます。
2. 設定済みテーブルをコラボレーションに関連付け、AWS Glue テーブルにアクセスできるサービスアクセスロールを提供します。
3. 設定されたモデルアルゴリズムの関連付けが設定されたテーブルにアクセスできるようにする [コラボレーション分析ルール](#) をテーブルに追加します。
4. モデルとデータを Clean Rooms ML で関連付けて設定すると、クエリを実行できるメンバーは SQL クエリを提供し、使用するモデルアルゴリズムを選択します。

モデルトレーニングが完了すると、そのメンバーはモデルトレーニングアーティファクトまたは推論結果のエクスポートを開始します。これらのアーティファクトまたは結果は、トレーニングされたモデル出力を受信できるメンバーに送信されます。結果レシーバーは、モデル出力を受け取る `MachineLearningConfiguration` 前に を設定する必要があります。

AWS Clean Rooms ML のモデル

AWS Clean Rooms ML は、2 つの当事者間でデータを共有しなくても、データ内の類似ユーザーを識別するためのプライバシー保護方法を提供します。最初のパーティはトレーニングデータを 持ち込み AWS Clean Rooms 、類似モデルを作成して設定し、コラボレーションに関連付けることができます。次に、トレーニングデータに似た類似セグメントを作成するために、シードデータがコラボレーションに持ち込まれます。

この動作の詳細な説明については、「[クロスアカウントジョブ](#)」を参照してください。

以下のトピックでは、Clean Rooms ML で AWS モデルを作成および設定する方法について説明します。

トピック

- [AWS Clean Rooms ML のプライバシー保護](#)
- [Clean Rooms ML のトレーニングデータ要件](#)
- [Clean Rooms ML のシードデータ要件](#)
- [AWS Clean Rooms ML モデル評価メトリクス](#)

AWS Clean Rooms ML のプライバシー保護

Clean Rooms ML は、トレーニングデータプロバイダーがシードデータに誰が含まれているかを知り、シードデータプロバイダーがトレーニングデータに誰が含まれているかを知ることができる、メ

メンバーシップ推論攻撃のリスクを軽減するように設計されています。この攻撃を防ぐためにいくつかの対策が講じられています。

まず、シードデータプロバイダーは Clean Rooms ML の出力を直接観察せず、トレーニングデータプロバイダーはシードデータを観察することができません。シードデータプロバイダーは、シードデータを出カセグメントに含めることもできます。

次に、トレーニングデータのランダムなサンプルから類似モデルを作成します。このサンプルには、シードオーディエンスと一致しない多数のユーザーが含まれています。このプロセスにより、ユーザーがデータになかったかどうかを判断することが難しくなります。これは、メンバーシップ推論のもう 1 つの手段です。

さらに、シード固有の類似モデルトレーニングの各パラメータに複数のシードカスタマーを使用できます。これにより、モデルがどれだけオーバーフィットできるかが制限され、ユーザーについてどれだけ推測できるかが制限されます。この結果、シードデータの最小サイズは 500 ユーザーとすることをお勧めします。

最後に、ユーザーレベルのメトリクスはトレーニングデータプロバイダーには提供されないため、メンバーシップ推論攻撃を受ける別の手段がなくなります。

Clean Rooms ML のトレーニングデータ要件

類似モデルを正常に作成するには、トレーニングデータが次の要件を満たしている必要があります。

- トレーニングデータは、Parquet、CSV、または JSON 形式である必要があります。

Note

Zstandard (ZSTD) 圧縮 Parquet データはサポートされていません。

- トレーニングデータはカタログ化する必要があります AWS Glue。詳細については、AWS Glue デベロッパーガイドの[AWS Glue データカタログの開始方法](#)を参照してください。スキーマは自動的に推測されるため、AWS Glue クローラを使用してテーブルを作成することをお勧めします。
- トレーニングデータとシードデータを含む Amazon S3 バケットは、他の Clean Rooms ML リソースと同じ AWS リージョンにあります。
- トレーニングデータには、少なくとも 100,000 個の一意のユーザー ID が含まれ、それぞれに少なくとも 2 つのアイテムインタラクションが含まれている必要があります。

- トレーニングデータには、少なくとも 100 万個のレコードが含まれている必要があります。
- [CreateTrainingDataset](#) アクションで指定されたスキーマは、AWS Glue テーブルの作成時に定義されたスキーマと一致する必要があります。
- 指定されたテーブルで定義されている必須フィールドは、[CreateTrainingDataset](#) アクションで定義されます。

フィールド タイプ	サポートさ れている データ型	必要	説明
USER_ID	string、int、bigint	はい	データセットでの各ユーザー用の一意の識別子。個人を特定できる情報 (PII) ではない値である必要があります。これは、ハッシュ化された ID または顧客 ID である場合があります。
ITEM_ID	string、int、bigint	はい	ユーザーが操作する各アイテムの一意の ID。
TIMESTAMP	bigint、int、timestamp	はい	ユーザーがアイテムを操作した

フィールド タイプ	サポートさ れている データ型	必要	説明
			時刻。値は Unix エポッ ク時刻形式 (秒単位) で ある必要が あります。

フィールド タイプ	サポートさ れている データ型	必要	説明
CATEGORICAL_FEATURE	string、int、float、bigint、double、boolean、array	いいえ	ユーザーまたはアイテムに関連するカテゴリデータをキャプチャします。これには、イベントタイプ (クリックや購入など)、ユーザーのデモグラフィック (年齢グループ、性別 – 匿名化)、ユーザーの所在地 (都市、国 – 匿名化)、アイテムカテゴリ (衣類や電子機器など)、アイテムのブランドなどが含まれます。

フィールド タイプ	サポートさ れている データ型	必要	説明
NUMERICAL _FEATURE	double、float、int、bigint	いいえ	ユーザーまたはアイテムに関連する数値データをキャプチャします。これには、ユーザーの購入履歴 (合計支払い額)、アイテムの価格、アイテムページへのアクセス回数、アイテムのユーザー評価などが含まれます。

- オプションで、最大 10 個のカテゴリ特徴量または数値特徴量を指定できます。

CSV 形式の有効なトレーニングデータセットの例を次に示します。

```

USER_ID,ITEM_ID,TIMESTAMP,EVENT_TYPE(CATEGORICAL FEATURE),EVENT_VALUE (NUMERICAL
FEATURE)
196,242,881250949,click,15
186,302,891717742,click,13
22,377,878887116,click,10
244,51,880606923,click,20
166,346,886397596,click,10

```

Clean Rooms ML のシードデータ要件

類似モデルのシードデータは、Amazon S3 バケットから直接取得することも、SQL クエリの結果から取得することもできます。

直接提供されるシードデータは、次の要件を満たしている必要があります。

- シードデータは、ユーザー IDs のリストを含む JSON 行形式である必要があります。
- シードサイズは 25~500,000 の一意のユーザー ID である必要があります。
- シードユーザーの最小数は、設定されたオーディエンスモデルの作成時に指定されたシードサイズの最小値と一致する必要があります。

以下は、CSV 形式の有効なトレーニングデータセットの例です。

```
{"user_id": "abc"}
{"user_id": "def"}
{"user_id": "ghijkl"}
{"user_id": "123"}
{"user_id": "456"}
{"user_id": "7890"}
```

AWS Clean Rooms ML モデル評価メトリクス

Clean Rooms ML はリコールおよび関連性スコアを計算してモデルのパフォーマンスを判断します。リコールは、類似データとトレーニングデータの類似性を比較します。関連性スコアは、モデルのパフォーマンスが良好かどうかではなく、オーディエンスの大きさを判断するために使用されます。

リコールは、類似セグメントがトレーニングデータとどの程度類似しているかを公平に測定するものです。リコールは、オーディエンス生成ジョブによってシードオーディエンスに含まれるトレーニングデータのサンプルからの、最も類似したユーザー (デフォルトでは最も類似した 20%) の割合です。値の範囲は 0~1 で、値が大きいほど対象者が良いことを示します。リコール値は、ビンの最大割合とほぼ等しい場合、オーディエンスモデルがランダム選択と同等であることを示します。

Clean Rooms ML はモデルの構築時に真陰性のユーザーを正確にラベル付けしていないため、AWS では、正確性、精度、F1 スコアよりも、これがより優れた評価メトリクスであると考えています。

セグメントレベルの関連性スコアは、-1 (最も類似しない) から 1 (最も類似する) までの値を持つ類似性の尺度です。Clean Rooms ML は、さまざまなセグメントサイズについて一連の関連性スコアを計算し、データに最適なセグメントサイズを決定するのに役立ちます。関連性スコアは、セグメント

サイズが大きくなるにつれて単調に減少するため、セグメントサイズが大きくなるにつれてシードデータに類似しなくなる可能性があります。セグメントレベルの関連性スコアが 0 に達すると、モデルは類似セグメントのすべてのユーザーがシードデータと同じディストリビューションに属すると予測します。出力サイズを大きくすると、類似セグメントに、シードデータと同じディストリビューションに属さないユーザーが含まれる可能性が高くなります。

関連性スコアは 1 つのキャンペーン内で標準化されるため、キャンペーン間の比較には使用しないでください。関連性スコアは、インベントリの品質、インベントリタイプ、広告のタイミングなど、関連性に加えて複数の複雑な要因によって影響を受けるため、ビジネス成果の単一ソースの証拠として使用しないでください。

関連性スコアはシードの品質を判断するためではなく、増減できるかどうかを判断するために使うべきです。次の例を考えます。

- すべて正のスコア – 類似していると予測される出力ユーザーの方が、類似セグメントに含まれるユーザーよりも多いことを示しています。これは、過去 1 か月間に歯磨き粉を購入したユーザーなど、大規模な市場に属するシードデータによく見られます。過去 1 か月に歯磨き粉を複数回購入したユーザーなど、比較的小さなシードデータを確認することをお勧めします。
- すべて負のスコア、または希望する類似セグメントサイズに対して負 – これは、Clean Rooms ML が、希望する類似セグメントサイズでは類似ユーザーが十分ではないと予測していることを示しています。これは、シードデータが具体的すぎるか、市場が小さすぎることが原因と考えられます。シードデータに適用するフィルターの数を減らすか、市場を拡大することをお勧めします。例えば、元のシードデータがベビーカーとチャイルドシートを購入した顧客だった場合、ベビー用品を複数購入した顧客に市場を拡大できます。

トレーニングデータプロバイダーは、関連性スコアを公開するかどうか、および関連性スコアを計算するバケットビンを決定します。

Clean Rooms ML のカスタムモデル

Clean Rooms ML を使用すると、コラボレーションのメンバーは Amazon ECR に保存されているドッカー化されたカスタムモデルアルゴリズムを使用して、データを共同で分析できます。これを行うには、モデルプロバイダーがイメージを作成し、Amazon ECR に保存する必要があります。[Amazon Elastic Container Registry ユーザーガイド](#)の手順に従って、カスタム ML モデルを含むプライベートリポジトリを作成します。

コラボレーションのメンバーは、適切なアクセス許可があれば、モデルプロバイダーになることができます。コラボレーションのすべてのメンバーは、モデルにデータを提供できます。このガイドで

は、データを提供するメンバーをデータプロバイダーと呼びます。コラボレーションを作成するメンバーはコラボレーション作成者であり、このメンバーはモデルプロバイダー、データプロバイダーの1つ、またはその両方です。

以下のトピックでは、カスタム ML モデルの作成に必要な情報について説明します。

トピック

- [カスタム ML モデリングの前提条件](#)
- [トレーニングコンテナのモデル作成ガイドライン](#)
- [推論コンテナのモデル作成ガイドライン](#)
- [モデルログとメトリクスの受信](#)

カスタム ML モデリングの前提条件

カスタム ML モデリングを実行する前に、次の点を考慮する必要があります。

- トレーニングされたモデルのモデルトレーニングと推論の両方をコラボレーションで実行するかどうかを決定します。
- 各コラボレーションメンバーが実行するロールを決定し、適切な機能を割り当てます。
 - モデルをトレーニングし、トレーニングされたモデルで推論を実行するメンバーに `CAN_QUERY` 機能を割り当てます。
 - コラボレーションの少なくとも1人のメンバー `CAN_RECEIVE_RESULTS` に を割り当てます。
 - トレーニング済みモデルのエクスポート `CAN_RECEIVE_MODEL_OUTPUT` または推論出力をそれぞれ受け取るメンバーに または `CAN_RECEIVE_INFERENCE_OUTPUT` の機能を割り当てます。ユースケースで必要な場合は、両方の機能を使用できます。
- エクスポートを許可するトレーニング済みモデルアーティファクトまたは推論結果の最大サイズを決定します。
- すべてのユーザーには、ロールに `CleanroomsFullAccess` および `CleanroomsMLFullAccess` ポリシーがアタッチされていることをお勧めします。カスタム ML モデルを使用するには、AWS Clean Rooms と AWS Clean Rooms ML SDKs の両方を使用する必要があります。
- IAM ロールに関する次の情報を考慮してください。
 - すべてのデータプロバイダーには、 が AWS Glue カタログとテーブル、および基盤となる Amazon S3 ロケーションからデータ AWS Clean Rooms を読み取ることができるサービスアクセスロールが必要です。これらのロールは、SQL クエリに必要なロールと似ています。これ

により、CreateConfiguredTableAssociationアクションを使用できます。詳細については、「[サービスロールを作成して、設定済みテーブルの関連付けを作成する](#)」を参照してください。

- メトリクスを受け取るすべてのメンバーには、CloudWatch メトリクスとログの書き込みを許可するサービスアクセスロールが必要です。このロールは、Clean Rooms ML がモデルトレーニングと推論 AWS アカウント 中に、すべてのモデルメトリクスとログをメンバーの に書き込むために使用されます。また、どのメンバーがメトリクスとログにアクセスできるかを判断するためのプライバシーコントロールも提供しています。これにより、CreateMLConfigurationアクションを使用できます。詳細については、[カスタム ML モデリングのサービスロールを作成する - ML 設定](#) を参照してください。

結果を受け取るメンバーは、Amazon S3 バケットに書き込むアクセス許可を持つサービスアクセスロールを提供する必要があります。このロールにより、Clean Rooms ML は結果 (トレーニング済みモデルアーティファクトまたは推論結果) を Amazon S3 バケットにエクスポートできます。これにより、CreateMLConfigurationアクションを使用できます。詳細については、「[カスタム ML モデリングのサービスロールを作成する - ML 設定](#)」を参照してください。

- モデルプロバイダーは、Amazon ECR リポジトリとイメージを読み取るアクセス許可を持つサービスアクセスロールを提供する必要があります。これにより、CreateConfigureModelAlgorithmアクションを使用できます。詳細については、「[カスタム ML モデルを提供するサービスロールを作成する](#)」を参照してください。
- トレーニングまたは推論用のデータセットを生成するMLInputChannelために を作成するメンバーは、Clean Rooms ML が SQL クエリを実行できるようにするサービスアクセスロールを提供する必要があります AWS Clean Rooms。これにより、CreateTrainedModelおよびStartTrainedModelInferenceJobアクションを使用できます。詳細については、「[データセットをクエリするサービスロールを作成する](#)」を参照してください。
- モデル作成者は、モデルの入力と出力が によって期待どおりに設定されていることを確認する[推論コンテナのモデル作成ガイドライン](#)ために、[トレーニングコンテナのモデル作成ガイドライン](#)と に従う必要があります AWS Clean Rooms。

トレーニングコンテナのモデル作成ガイドライン

このセクションでは、Clean Rooms ML のカスタム ML モデルアルゴリズムを作成するときにモデルプロバイダーが従うべきガイドラインについて詳しく説明します。

- SageMaker AI [「デベロッパーガイド」](#)の説明に従って、適切な SageMaker AI トレーニング対応コンテナベースイメージを使用します。次のコードでは、サポートされているコンテナベースイメージをパブリック SageMaker AI エンドポイントからプルできます。

```
ecr_registry_endpoint='763104351884.dkr.ecr.$REGION.amazonaws.com'
base_image='pytorch-training:2.3.0-cpu-py311-ubuntu20.04-sagemaker'
aws ecr get-login-password --region $REGION | docker login --username AWS --password-stdin $ecr_registry_endpoint
docker pull $ecr_registry_endpoint/$base_image
```

- モデルをローカルで作成する場合は、モデルをローカル、開発インスタンス、の SageMaker AI トレーニング、および Clean Rooms ML でテストできるように AWS アカウント、以下を確認してください。
- さまざまな環境変数を使用して、トレーニング環境に関する有用なプロパティにアクセスするトレーニングスクリプトを作成することをお勧めします。Clean Rooms ML は、次の引数を使用してモデルコードのトレーニングを呼び出します: SM_MODEL_DIR、SM_OUTPUT_DIR、SM_CHANNEL_TRAIN、FILE_FORMAT。これらのデフォルトは、Clean Rooms ML によって使用され、すべての関係者のデータを使用して独自の実行環境で ML モデルをトレーニングします。
- Clean Rooms ML は、Docker コンテナの/opt/ml/input/data/*channel-name*ディレクトリを介してトレーニング入力チャネルを利用できるようにします。各 ML 入力チャネルは、CreateTrainedModelリクエストでchannel_name提供された対応するチャネルに基づいてマッピングされます。

```
parser = argparse.ArgumentParser()# Data, model, and output directories

parser.add_argument('--model_dir', type=str, default=os.environ.get('SM_MODEL_DIR', "/opt/ml/model"))
parser.add_argument('--output_dir', type=str, default=os.environ.get('SM_OUTPUT_DIR', "/opt/ml/output/data"))
parser.add_argument('--train_dir', type=str, default=os.environ.get('SM_CHANNEL_TRAIN', "/opt/ml/input/data/train"))
parser.add_argument('--train_file_format', type=str, default=os.environ.get('FILE_FORMAT', "csv"))
```

- モデルコードで使用される共同作業者のスキーマに基づいて合成データセットまたはテストデータセットを生成できることを確認します。
- モデルアルゴリズム AWS Clean Rooms をコラボレーションに関連付ける AWS アカウント 前に、SageMaker AI トレーニングジョブを自分で実行できることを確認してください。

次のコードには、ローカルテスト、SageMaker AI トレーニング環境テスト、Clean Rooms ML と互換性のあるサンプル Docker ファイルが含まれています。

```
FROM 763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-training:2.3.0-cpu-
py311-ubuntu20.04-sagemaker
MAINTAINER $author_name

ENV PYTHONDONTWRITEBYTECODE=1 \
    PYTHONUNBUFFERED=1 \
    LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:/usr/local/lib"

ENV PATH="/opt/ml/code:${PATH}"

# this environment variable is used by the SageMaker PyTorch container to determine
# our user code directory
ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code

# copy the training script inside the container
COPY train.py /opt/ml/code/train.py
# define train.py as the script entry point
ENV SAGEMAKER_PROGRAM train.py
ENTRYPOINT ["python", "/opt/ml/code/train.py"]
```

- コンテナの障害を最適にモニタリングするには、障害の理由でログをエクスポートし、デバッグすることをお勧めします。GetTrainedModel レスポンスでは、Clean Rooms ML は、このファイルの最初の 1024 文字を返しますStatusDetails。
- モデルの変更を完了し、SageMaker AI 環境でテストする準備ができたなら、指定された順序で次のコマンドを実行します。

```
export ACCOUNT_ID=xxx
export REPO_NAME=xxx
export REPO_TAG=xxx
export REGION=xxx

docker build -t $ACCOUNT_ID.dkr.ecr.us-west-2.amazonaws.com/$REPO_NAME:$REPO_TAG

# Sign into AWS $ACCOUNT_ID/ Run aws configure
# Check the account and make sure it is the correct role/credentials
aws sts get-caller-identity
aws ecr create-repository --repository-name $REPO_NAME --region $REGION
aws ecr describe-repositories --repository-name $REPO_NAME --region $REGION
```

```
# Authenticate Docker
aws ecr get-login-password --region $REGION | docker login --username AWS --password-
stdin $ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com

# Push To ECR Images
docker push $ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com$REPO_NAME:$REPO_TAG

# Create Sagemaker Training job
# Configure the training_job.json with
# 1. TrainingImage
# 2. Input DataConfig
# 3. Output DataConfig
aws sagemaker create-training-job --cli-input-json file://training_job.json --region
$REGION
```

SageMaker AI ジョブが完了し、モデルアルゴリズムに満足したら、Amazon ECR レジストリを AWS Clean Rooms ML に登録できます。CreateConfiguredModelAlgorithm アクションを使用してモデルアルゴリズムを登録し、CreateConfiguredModelAlgorithmAssociationを使用してコラボレーションに関連付けます。

推論コンテナのモデル作成ガイドライン

このセクションでは、Clean Rooms ML の推論アルゴリズムを作成するときにモデルプロバイダーが従うべきガイドラインについて詳しく説明します。

- SageMaker AI [デベロッパーガイド](#) の説明に従って、適切な SageMaker AI 推論がサポートするコンテナベースイメージを使用します。次のコードでは、サポートされているコンテナベースイメージをパブリック SageMaker AI エンドポイントからプルできます。

```
ecr_registry_endpoint='763104351884.dkr.ecr.$REGION.amazonaws.com'
base_image='pytorch-inference:2.3.0-cpu-py311-ubuntu20.04-sagemaker'
aws ecr get-login-password --region $REGION | docker login --username AWS --password-
stdin $ecr_registry_endpoint
docker pull $ecr_registry_endpoint/$base_image
```

- モデルをローカルで作成する場合は、モデルをローカル、開発インスタンス、の SageMaker AI バッチ変換、および Clean Rooms ML でテストできるように AWS アカウント、以下を確認してください。

- Clean Rooms ML は、推論からのモデルアーティファクトを docker コンテナの /opt/ml/model ディレクトリを介して推論コードで使えるようにします。
- Clean Rooms ML は入力を行ごとに分割し、MultiRecordバッチ戦略を使用して、変換されたすべてのレコードの最後に改行文字を追加します。
- モデルコードで使用される共同作業者のスキーマに基づいて、合成またはテスト推論データセットを生成できることを確認します。
- モデルアルゴリズム AWS Clean Rooms をコラボレーションに関連付ける AWS アカウント 前に、SageMaker AI バッチ変換ジョブを自分で実行できることを確認してください。

次のコードには、ローカルテスト、SageMaker AI 変換環境テスト、Clean Rooms ML と互換性のあるサンプル Docker ファイルが含まれています。

```
FROM 763104351884.dkr.ecr.us-east-1.amazonaws.com/pytorch-inference:1.12.1-cpu-py38-ubuntu20.04-sagemaker

ENV PYTHONUNBUFFERED=1

COPY serve.py /opt/ml/code/serve.py
COPY inference_handler.py /opt/ml/code/inference_handler.py
COPY handler_service.py /opt/ml/code/handler_service.py
COPY model.py /opt/ml/code/model.py

RUN chmod +x /opt/ml/code/serve.py

ENTRYPOINT ["/opt/ml/code/serve.py"]
```

- モデルの変更を完了し、SageMaker AI 環境でテストする準備ができれば、指定された順序で次のコマンドを実行します。

```
export ACCOUNT_ID=xxx
export REPO_NAME=xxx
export REPO_TAG=xxx
export REGION=xxx

docker build -t $ACCOUNT_ID.dkr.ecr.us-west-2.amazonaws.com/$REPO_NAME:$REPO_TAG

# Sign into AWS $ACCOUNT_ID/ Run aws configure
# Check the account and make sure it is the correct role/credentials
aws sts get-caller-identity
aws ecr create-repository --repository-name $REPO_NAME --region $REGION
```

```
aws ecr describe-repositories --repository-name $REPO_NAME --region $REGION

# Authenticate Docker
aws ecr get-login-password --region $REGION | docker login --username AWS --password-
stdin $ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com

# Push To ECR Repository
docker push $ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com$REPO_NAME:$REPO_TAG

# Create Sagemaker Model
# Configure the create_model.json with
# 1. Primary container -
#   a. ModelDataUrl - S3 Uri of the model.tar from your training job
aws sagemaker create-model --cli-input-json file://create_model.json --region $REGION

# Create Sagemaker Transform Job
# Configure the transform_job.json with
# 1. Model created in the step above
# 2. MultiRecord batch strategy
# 3. Line SplitType for TransformInput
# 4. AssembleWith Line for TransformOutput
aws sagemaker create-transform-job --cli-input-json file://transform_job.json --
region $REGION
```

SageMaker AI ジョブが完了し、バッチ変換に満足したら、Amazon ECR レジストリを AWS Clean Rooms ML に登録できます。CreateConfiguredModelAlgorithm アクションを使用してモデルアルゴリズムを登録し、CreateConfiguredModelAlgorithmAssociationを使用してコラボレーションに関連付けます。

モデルログとメトリクスの受信

カスタムモデルトレーニングまたは推論からログとメトリクスを受信するには、メンバーが必要な CloudWatch アクセス許可を提供する有効なロールを持つ [ML 設定を作成](#)しておく必要があります (「[カスタム ML モデリングのサービスロールの作成 - ML 設定](#)」を参照)。

システムメトリクス

CPU 使用率やメモリ使用率など、トレーニングと推論の両方のシステムメトリクスは、有効な ML 設定を持つコラボレーションのすべてのメンバーに発行されます。これらのメトリクスは、それぞれ /aws/cleanroomsm1/TrainedModels または /aws/cleanroomsm1/

TrainedModelInferenceJobs 名前空間の CloudWatch Metrics を介してジョブの進行状況として表示できます。

モデルログ

モデルログへのアクセスは、設定された各モデルアルゴリズムのプライバシー設定ポリシーによって提供されます。モデル作成者は、設定されたモデルアルゴリズムを (コンソールまたは CreateConfiguredModelAlgorithmAssociation API を介して) コラボレーションに関連付けるときに、プライバシー設定ポリシーを設定します。プライバシー設定ポリシーを設定すると、どのメンバーがモデルログを受信できるかが制御されます。

さらに、モデル作成者は、プライバシー設定ポリシーでフィルターパターンを設定して、ログイベントをフィルタリングできます。モデルコンテナが stdout または stderr に送信し、フィルターパターン (設定されている場合) に一致するすべてのログが Amazon CloudWatch Logs に送信されます。モデルログは /aws/cleanroomsm1/TrainedModelInferenceJobs、CloudWatch ロググループ /aws/cleanroomsm1/TrainedModels または でそれぞれ使用できます。

カスタム定義メトリクス

モデルアルゴリズムを設定する場合 (コンソールまたは CreateConfiguredModelAlgorithm API 経由)、モデル作成者は、出力ログで検索する特定のメトリクス名と正規表現ステートメントを指定できます。これらは、/aws/cleanroomsm1/TrainedModels 名前空間の CloudWatch Metrics を介してジョブの進行状況として表示できます。設定されたモデルアルゴリズムに関連付ける場合、モデル作成者はメトリクスのプライバシー設定でオプションのノイズレベルを設定して、カスタムメトリクスの傾向を可視化しながら raw データの出力を回避できます。ノイズレベルが設定されている場合、メトリクスはリアルタイムではなくジョブの最後に発行されます。

Cryptographic Computing for Clean Rooms

Cryptographic Computing for Clean Rooms (C3R) は、[分析ルール](#)に加えて AWS Clean Rooms 使用できるの機能です。C3R を使うと、組織は機密データをまとめて、データ分析から新しいインサイトを引き出せると同時に、プロセスの中で関係者が知ることのできる事項を暗号化によって制限できます。C3R は、コラボレーションで機密データを扱いたい、クラウド内の暗号化されたデータのみを使用する必要がある複数の関係者が使用できます。

C3R 暗号化クライアントは、で使用するデータを[暗号化](#)するために使用できるクライアント側の暗号化ツールです AWS Clean Rooms。C3R 暗号化クライアントを使用する場合、AWS Clean Rooms データはコラボレーションで使用中も暗号化によって保護されます。通常の AWS Clean Rooms コラボレーションと同様に、入力データはリレーショナルデータベースであり、

計算は SQL クエリとして表されます。ただし、暗号化されたデータに対する SQL クエリについて、C3R は一部の限られたクエリしかサポートしていません。

具体的には、暗号で保護されたデータに対する SQL JOIN および SELECT のステートメントをサポートしています。入力テーブルの各列は、次の SQL ステートメントタイプのいずれか 1 つだけで使用できます。

- JOIN ステートメントで使用される、暗号化によって保護された列は `fingerprint` 列と呼ばれます。
- SELECT ステートメントで使用される、暗号化によって保護された列は `sealed` 列と呼ばれます。
- JOIN または SELECT ステートメントで使用される、暗号化によって保護されていない列は `cleartext` 列と呼ばれます。

場合によっては、GROUP BY ステートメントが `fingerprint` 列でサポートされることもあります。詳細については、「[Fingerprint 列](#)」を参照してください。現在 C3R では、関連する分析ルールで許可されている場合でも、WHERE 句、または SUM や AVERAGE といった集約関数など、他の SQL 構文の暗号化済みデータへの使用はサポートしていません。

C3R はテーブルの個々のセル内のデータを保護するように設計されています。C3R のデフォルト設定を使用すると、お客様がコラボレーションを通じてサードパーティに提供する基データは、コンテンツが AWS Clean Rooms 内部で使用されている間は暗号化されたままになります。C3R では、すべての `sealed` 列に業界標準の AES-GCM 暗号化を使用し、`fingerprint` 列の保護には Hash-based Message Authentication Code (HMAC) と呼ばれる業界標準の疑似ランダム機能を使用します。

C3R はテーブル内のデータを暗号化しますが、以下の情報は推測できる場合があります。

- テーブルの列数、列名、行数など、テーブル自体に関する情報。
- ほとんどの標準的な暗号化形式と同様、C3R では、暗号化された値の長さについては隠蔽を試みません。C3R には、暗号化された値をパディングしてクリアテキストの正確な長さを隠蔽する機能がありますが、各列のクリアテキストの長さの上限は、第三者に把握される可能性があります。
- 暗号化された C3R テーブルに特定の行がいつ追加されたかなど、ログレベルの情報。

C3R の詳細については、以下のトピックを参照してください。

トピック

- [Cryptographic Computing for Clean Rooms を使用する際の考慮事項](#)
- [Cryptographic Computing for Clean Rooms でサポートされているファイルとデータの種類](#)
- [Cryptographic Computing for Clean Rooms での列名](#)

- [Cryptographic Computing for Clean Rooms での列タイプ](#)
- [暗号コンピューティングパラメータ](#)
- [Cryptographic Computing for Clean Rooms のオプションフラグ](#)
- [クエリと Cryptographic Computing for Clean Rooms](#)
- [C3R 暗号化クライアントのガイドライン](#)

Cryptographic Computing for Clean Rooms を使用する際の考慮事項

Cryptographic Computing for Clean Rooms (C3R) は、データ保護を最大限に強化することを目的としています。ただし、一部のユースケースでは、追加機能と引き換えにデータの保護レベルを下げることでメリットが得られる場合があります。こうした特定のトレードオフは、C3R を最も安全な設定から変更することで実現できます。ユーザーは、これらのトレードオフを認識し、それが自身のユースケースに適しているかどうかを判断する必要があります。考慮すべきトレードオフは次のとおりです。

トピック

- [テーブル内でcleartextデータと暗号化データの混在を許可する](#)
- [fingerprint列で値の繰り返しを許可する](#)
- [fingerprint列の命名方法に関する制限を緩和する](#)
- [NULL 値の表現方法を決定する](#)

これらのシナリオでのパラメータの使用方法の詳細については、「[暗号コンピューティングパラメータ](#)」を参照してください。

テーブル内でcleartextデータと暗号化データの混在を許可する

すべてのデータをクライアント側で暗号化することで、最大限のデータ保護が可能になります。ただし、これにより特定の種類のクエリ (SUM 集約関数など) が制限されます。cleartextデータを許可することのリスクは、暗号化されたテーブルにアクセスできる人なら誰でも暗号化された値に関する情報を推測できるようになることです。これは、cleartextおよび関連するデータを統計的に分析することで可能になります。

例えば、City と State の列があるとします。City 列はcleartextで、State 列は暗号化されています。City 列の値 Chicago を見れば、State の値がIllinoisであることを高い確率で特定できます。逆に、一方の列が City で、もう一方の列が EmailAddress の場合、暗号化されている EmailAddress の情報がcleartext City によって明らかになることはありません。

このシナリオのパラメータの詳細については、「[\[cleartext 列を許可\] パラメータ](#)」を参照してください。

fingerprint列で値の繰り返しを許可する

最も安全な方法では、どのfingerprint列にも変数のインスタンスが 1 つだけ含まれていると想定されています。1 つのfingerprint列で項目を繰り返すことはできません。C3R 暗号化クライアントは、これらのcleartext値をランダム値と見分けがつかない一意の値にマッピングします。したがって、これらのランダム値からcleartextに関する情報を推測することはできません。

fingerprint列で値が繰り返される場合のリスクは、値が繰り返されるとランダムに見える値も繰り返されることです。したがって理論的には、暗号化されたテーブルにアクセスできる人なら誰でもfingerprint列の統計分析を行って、cleartext値に関する情報を明らかにできる可能性があります。

例えば、fingerprint列が State で、テーブルのすべての行が米国の世帯に対応しているとします。頻度分析を行うことで、どの州が California でどの州が Wyoming であるかを高い確率で推測できます。この推測が可能なのは、California の方が Wyoming よりも住民の数がはるかに多いからです。逆に、fingerprint列が世帯識別子に関するもので、数百万件のエントリからなるデータベースの中で各世帯が 1 ~ 4 回出現したとします。この場合、頻度分析によって有用な情報が明らかになることはまずありません。

このシナリオのパラメータの詳細については、「[\[複製を許可\] パラメータ](#)」を参照してください。

fingerprint列の命名方法に関する制限を緩和する

デフォルトでは、暗号化されたfingerprint列を使用して 2 つのテーブルが結合される場合、各テーブルのそれらの列は名前が同じであると想定されます。この結果の技術的な理由は、デフォルトでは、各fingerprint列を暗号化するために異なる暗号キーを派生させるからです。このキーは、コラボレーションの共有シークレットキーと列名の組み合わせから派生します。列名の異なる 2 つの列を結合しようとする、異なるキーが派生し、有効な結合を処理できません。

この問題に対処するには、各列名からキーを派生させる機能をオフにします。すると、C3R 暗号化クライアントはすべてのfingerprint列に 1 つの派生キーを使用します。リスクは、別の種類の頻度分析を行うと、情報が明らかになる可能性があることです。

もう一度 City と State の例を使ってみましょう。各fingerprint列で同じランダム値を派生させる (列名を組み込まない) 場合、New York では City 列と State 列のランダム値が同じになります。ニューヨークは、米国でも数少ない、City と State の名前が同じ都市の 1 つです。逆に、データセットの各列の値がまったく異なれば、情報が漏れることはありません。

このシナリオのパラメータの詳細については、「[\[名前の異なる列の JOIN を許可\] パラメータ](#)」を参照してください。

NULL 値の表現方法を決定する

選択肢は、NULL 値を他の値と同様に暗号化処理 (暗号化と HMAC) するかどうかです。NULL 値を他の値と同様に処理しないと、情報が漏洩する可能性があります。

例えば、Middle Name 列のcleartextの NULL が、ミドルネームのない人を示しているとしします。これらの値を暗号化しないと、暗号化されたテーブルのどの行がミドルネームを持たない人に使われているかが漏れてしまいます。そうした情報は、一部の集団の一部の人々にとっては、ある種の識別信号となる可能性があります。しかし、NULL 値を暗号化処理すると、特定の SQL クエリが異なる動作になります。例えば GROUP BY 句では、fingerprint列内のfingerprint NULL 値がまとめてグループ化されなくなります。

このシナリオのパラメータの詳細については、「[\[NULL 値を保存\] パラメータ](#)」を参照してください。

Cryptographic Computing for Clean Rooms でサポートされているファイルとデータの種類

C3R 暗号化クライアントは、以下の種類のファイルを認識します。

- CSV ファイル
- Parquet ファイル

C3R 暗号化クライアントの `--fileFormat` フラグを使用して、ファイル形式を明示的に指定できます。明示的に指定した場合、ファイル形式はファイル拡張子によって判断されません。

トピック

- [CSV ファイル](#)
- [Parquet ファイル](#)
- [文字列以外の値の暗号化](#)

CSV ファイル

.csv 拡張子の付いたファイルは CSV 形式で、UTF-8 でエンコードされたテキストを含むものとみなされます。C3R 暗号化クライアントはすべての値を文字列として扱います。

.csv ファイルでサポートされるプロパティ

C3R 暗号化クライアントでは、.csv ファイルに次のプロパティが必要です。

- 各列に一意の名前を付ける最初のヘッダー行 (含まれる場合と含まれない場合があります)。
- カンマ区切り (現在、カスタム区切り文字はサポートされていません)。
- UTF-8 でエンコードされたテキスト。

.csv エントリからの空白の削除

.csv エントリから先頭および末尾の空白が両方とも削除されます。

.csv ファイルのカスタム NULL エンコーディング

.csv ファイルではカスタム NULL エンコーディングを使用できます。

C3R 暗号化クライアントでは、`--csvInputNULLValue=<csv-input-null>` フラグを使用して入力データの NULL エントリにカスタムエンコーディングを指定できます。C3R 暗号化クライアントは、`--csvOutputNULLValue=<csv-output-null>` フラグを使用することで、生成された出力ファイル内の NULL エントリに対してカスタムエンコーディングを使用できます。

Note

特に SQL テーブルのようなよりリッチな表形式のコンテキストでは、NULL エントリは欠落したコンテンツとみなされます。.csv は歴史的な理由からこの特性を明示的にサポートしていませんが、空白だけを含む空のエントリは NULL と見なすのが一般的な慣習です。したがって、これは C3R 暗号化クライアントのデフォルト動作であり、必要に応じてカスタマイズできます。

C3R による .csv エントリの解釈

次の表は、`--csvInputNULLValue=<csv-input-null>` および `--csvOutputNULLValue=<csv-output-null>` のフラグに指定された値 (存在する場合) に基づいて、.csv エントリを (わかりやすくするために cleartext から cleartext に) 整列化する方法の例を示しています。引用符の外側にある先頭と末尾の空白は、C3R が値の意味を解釈する前に削除されます。

<csv-input-null>	<csv-output-null>	入力エントリ	出力エントリ
None	None	,#####,	,#####,
None	None	, ##### ,	,#####,
None	None	, "#####",	,#####,
None	None	, "#####" ,	,#####,
None	None	, ,	, ,
None	None	, ,	, ,
None	None	, "",	, ,
None	None	, " ",	, " ",
None	None	, " " ,	, " ",
"#####"	"NULL"	,#####,	,NULL,
"#####"	"NULL"	, ##### ,	,NULL,
"#####"	"NULL"	, "#####",	,NULL,
"#####"	"NULL"	, "#####" ,	,NULL,
None	"NULL"	, ,	,NULL,
None	"NULL"	, ,	,NULL,
None	"NULL"	, "",	,NULL,
None	"NULL"	, " ",	, " ",
None	"NULL"	, " " ,	, " ",
""	"NULL"	, ,	,NULL,
""	"NULL"	, ,	,NULL,

<csv-input-null>	<csv-output-null>	入力エントリ	出力エントリ
""	"NULL"	, "",	, "",
""	"NULL"	, " ",	, " ",
""	"NULL"	, " " ,	, " " ,
"\""	"NULL"	, ,	, ,
"\""	"NULL"	, ,	, ,
"\""	"NULL"	, "",	, NULL,
"\""	"NULL"	, " ",	, " ",
"\""	"NULL"	, " " ,	, " " ,

ヘッダーのない CSV ファイル

ソースの .csv ファイルでは、各列に一意の名前を付けるヘッダーが最初の行にななくてもかまいません。ただし、ヘッダー行のない .csv ファイルには位置暗号化スキーマが必要です。ヘッダー行のある .csv ファイルと Parquet ファイルの両方に使用される一般的なマッピングスキーマの代わりに、位置暗号化スキーマが必要になります。

位置暗号化スキーマは、出力列を名前ではなく位置で指定します。マッピング暗号化スキーマは、ソースの列名をターゲットの列名にマッピングします。両方のスキーマ形式の詳細な説明や例については、「[マッピングテーブルスキーマと位置テーブルスキーマ](#)」を参照してください。

Parquet ファイル

.parquet 拡張子の付いたファイルは、Apache Parquet 形式であるとみなされます。

サポートされている Parquet データ型

C3R 暗号化クライアントでは、AWS Clean Roomsでサポートされているデータ型を表す Parquet ファイル内の複雑でない (つまり、プリミティブ型の) データを処理できます。

ただし、sealed列には文字列の列しか使用できません。

以下の Parquet データ型 がサポートされています。

- 以下の論理アノテーション付きの Binary プリミティブ型
 - `--parquetBinaryAsString` が設定されている場合は不要 (STRING データ型)
 - `Decimal(scale, precision)` (DECIMAL データ型)
 - `String` (STRING データ型)
- 論理アノテーションのない Boolean プリミティブデータ型 (BOOLEAN データ型)
- 論理アノテーションのない Double プリミティブデータ型 (DOUBLE データ型)
- `Decimal(scale, precision)` 論理アノテーション付きの `Fixed_Len_Binary_Array` プリミティブ型 (DECIMAL データ型)
- 論理アノテーションのない Float プリミティブデータ型 (FLOAT データ型)
- 以下の論理アノテーション付きの Int32 プリミティブ型
 - 不要 (INT データ型)
 - `Date` (DATE データ型)
 - `Decimal(scale, precision)` (DECIMAL データ型)
 - `Int(16, true)` (SMALLINT データ型)
 - `Int(32, true)` (INT データ型)
- 以下の論理アノテーション付きの Int64 プリミティブデータ型
 - 不要 (BIGINT データ型)
 - `Decimal(scale, precision)` (DECIMAL データ型)
 - `Int(64, true)` (BIGINT データ型)
 - `Timestamp(isUTCAdjusted, TimeUnit.MILLIS)` (TIMESTAMP データ型)
 - `Timestamp(isUTCAdjusted, TimeUnit.MICROS)` (TIMESTAMP データ型)
 - `Timestamp(isUTCAdjusted, TimeUnit.NANOS)` (TIMESTAMP データ型)

文字列以外の値の暗号化

現在、sealed列では文字列値のみがサポートされています。

.csv ファイルの場合、C3R 暗号化クライアントはすべての値を UTF-8 でエンコードされたテキストとして扱い、暗号化前にそれらを異なる方法で解釈しようとはしません。

フィンガープリント列では、データ型は等価クラスにグループ化されます。等価クラスは、代表的なデータ型を使用して同等かどうかを明確に比較できるデータ型のセットです。

等価クラスを使用すると、元の表現に関係なく、同一のフィンガープリントを同じセマンティック値に割り当てることができます。ただし、2つの等価クラスで同じ値があっても、同じフィンガープリント列にはなりません。

例えば、42 の INTEGRAL 値には、元の値が SMALLINT、INT、または BIGINT であったかどうかにかかわらず、同じフィンガープリントが割り当てられます。また、0 の INTEGRAL 値が FALSE の BOOLEAN 値 (値 0 で表される) と一致することはありません。

フィンガープリント列では、次の等価クラスと対応する AWS Clean Rooms データ型がサポートされています。

等価クラス	サポートされている AWS Clean Rooms データ型
BOOLEAN	BOOLEAN
DATE	DATE
INTEGRAL	BIGINT, INT, SMALLINT
STRING	CHAR, STRING, VARCHAR

Cryptographic Computing for Clean Rooms での列名

デフォルトでは、Cryptographic Computing for Clean Rooms においては列の名前が重要になります。

[名前異なる列の JOIN を許可] パラメータの値が [false] の場合は、fingerprint列の暗号化時に列名が使用されます。このためデフォルトでは、コラボレーターが事前に調整して、クエリで JOIN ステートメントを使用するデータで同じターゲット列名を使用する必要があります。デフォルトでは、JOIN のために暗号化された名前異なる列は、どの値でも JOIN が正常に処理されません。

[名前異なる列の JOIN を許可] パラメータの値が [true] の場合、fingerprint列として暗号化された複数の列にわたる JOIN ステートメントは成功します。このパラメータを使用してデータを暗号化すると、cleartext値をある程度推測できる可能性があります。例えば、ある行の City 列と State 列の両方に同じ Hash-based Message Authentication Code (HMAC) 値がある場合、その値は New York である可能性があります。

列ヘッダー名の正規化

列ヘッダー名は C3R 暗号化クライアントによって正規化されます。変換後の出力では、先頭と末尾のスペースはすべて削除され、列名は小文字になります。

正規化は、列名の影響を受ける可能性のある他のすべての計算や操作の前に適用されます。出力されるファイルには、正規化された名前のみが含まれます。

Cryptographic Computing for Clean Rooms での列タイプ

このトピックでは、Cryptographic Computing for Clean Rooms での列タイプに関する情報を提供します。

トピック

- [Fingerprint列](#)
- [シール列](#)
- [Cleartext列](#)

Fingerprint列

Fingerprint列は、JOIN ステートメントで使用される、暗号化によって保護された列です。

fingerprint列のデータを復号化することはできません。復号化できるのは、シール列のデータだけです。

Fingerprint列は次の SQL 句と関数でのみ使用する必要があります。

- 他のfingerprint列に対する JOIN (INNER, OUTER, LEFT, RIGHT, or FULL)
 - allowJoinsOnColumnsWithDifferentNames パラメータの値が false に設定されている場合、JOIN の両方のfingerprint列が同じ名前であることも必要になります。
- SELECT COUNT()
- SELECT COUNT(DISTINCT)
- GROUP BY (コラボレーションで preserveNulls パラメータの値が true に設定されている場合にのみ使用)

これらの制約に違反するクエリは、誤った結果をもたらす可能性があります。

シール列

シール列は、SELECT ステートメントで使用される、暗号化によって保護された列です。

シール列は、次の SQL 句と関数でのみ使用する必要があります。

- SELECT
- SELECT ... AS
- SELECT COUNT()

Note

SELECT COUNT(DISTINCT) はサポートされていません。

これらの制約に違反するクエリは、誤った結果をもたらす可能性があります。

暗号化前のsealed列のデータのパディング

列をsealed列にするように指定すると、C3R からどの種類のパディングを選択するかをたずねられます。暗号化前のデータのパディングは任意です。パディングを使用しない場合 (パディングタイプ none) は、暗号化されたデータの長さがcleartextのサイズを示します。状況によっては、cleartextのサイズによってプレーンテキストが明らかになる場合もあります。パディングを使用する場合 (パディングタイプ fixed または max) は、まずすべての値が共通のサイズにパディングされ、次に暗号化されます。パディングを使用すると、暗号化されたデータの長さによって、サイズの上限は示されるものの、元のcleartextの長さに関するそれ以外の情報は得られません。

特定の列のパディングが必要で、その列のデータの最大バイト長がわかっている場合は、fixed パディングを使用し、少なくともその列の最大バイト長と同じ大きさの length 値を使用してください。

Note

値が指定した length 値より長いとエラーが発生し、暗号化は失敗します。

列のパディングが必要で、その列のデータの最大バイト長がわかっていない場合は、max パディングを使用します。このパディングモードは、すべてのデータを、最長値に追加の length バイトを加えた長さにパディングします。

 Note

データをまとめて暗号化したり、テーブルを新しいデータで定期的に更新したりする場合、max パディングを行うと、指定したバッチ内の最も長いプレーンテキストエントリの長さ (プラス length バイト) までエントリがパディングされることに注意してください。つまり、暗号文の長さはバッチごとに異なる可能性があります。したがって、列の最大バイト長がわかっている場合は、max の代わりに fixed 使用してください。

Cleartext列

Cleartext 列は、JOIN または SELECT ステートメントで使用される、暗号によって保護されていない列です。

Cleartext列は SQL クエリのどの部分でも使用できます。

暗号コンピューティングパラメータ

暗号コンピューティングパラメータは、[コラボレーションの作成](#)時に、Cryptographic Computing for Clean Rooms (C3R) を使用してコラボレーションに指定できます。AWS Clean Rooms コンソールまたは CreateCollaboration API オペレーションを使用してコラボレーションを作成できます。コンソールでは、{暗号コンピューティングをサポート} オプションをオンにすると、[暗号コンピューティングパラメータ] のパラメータ値を設定できます。詳細については、以下のトピックを参照してください。

トピック

- [\[cleartext 列を許可\] パラメータ](#)
- [\[複製を許可\] パラメータ](#)
- [\[名前の異なる列の JOIN を許可\] パラメータ](#)
- [\[NULL 値を保存\] パラメータ](#)

[cleartext 列を許可] パラメータ

コンソールでは、[コラボレーションの作成](#)時に [cleartext 列を許可] パラメータを設定して、暗号化されたデータを含むテーブルで cleartext データを許可するかどうかを指定できます。

次の表で、[cleartext 列を許可] パラメータの値について説明します。

パラメータ値	説明
いいえ	暗号化されたテーブルでCleartext 列を使用できません。すべてのデータは暗号で保護されます。
あり	暗号化されたテーブルでCleartext 列を使用できます。 Cleartext 列は暗号化によって保護されず、cleartext として含まれます。行の cleartext データによって、テーブル内の他のデータについて明らかになるおそれがある情報をメモしておく必要があります。 特定の列で SUM または AVG を実行するには、その列が cleartext である必要があります。

CreateCollaboration API オペレーションを使用して、dataEncryptionMetadata パラメータの allowCleartext 値を true または false に設定できます。API オペレーションの詳細については、「[AWS Clean Rooms API リファレンス](#)」を参照してください。

Cleartext 列は、テーブル固有のスキーマで cleartext に分類される列に対応します。これらの列のデータは暗号化されておらず、どのような方法でも使用できます。Cleartext 列は、データの機密性が低い場合や、暗号化された sealed 列や fingerprint 列で許容される以上の柔軟性が必要な場合に役立ちます。

[複製を許可] パラメータ

コンソールでは、[コラボレーションの作成](#)時に [複製を許可] パラメータを設定して、暗号化された列の JOIN クエリで NULL 値以外の重複する値を含めるかどうかを指定できます。

Important

[複製を許可]、[\[名前が異なる列の JOIN を許可\]](#)、[\[NULL 値を保存\]](#) の各パラメータには、別々でありながら関連する効果があります。

次の表で、[複製を許可] パラメータの値について説明します。

パラメータ値	説明
いいえ	1 つの fingerprint 列で値の繰り返しは許可されません。1 つの fingerprint 列の値は、すべて一意でなければなりません。
あり	1 つの fingerprint 列で値の繰り返しが許可されます。 値が繰り返される列を結合する必要がある場合は、この値を [はい] に設定します。[はい] に設定すると、C3R テーブルまたは結果の fingerprint 列に示される頻度パターンによって、cleartext データの構造に関するその他の情報が推察可能になるおそれがあります。

CreateCollaboration API オペレーションを使用して、dataEncryptionMetadata パラメータの allowDuplicates 値を true または false に設定できます。API オペレーションの詳細については、「[AWS Clean Rooms API リファレンス](#)」を参照してください。

デフォルトでは、暗号化されたデータを JOIN クエリで使用する必要がある場合、C3R 暗号化クライアントでは、それらの列に重複する値がないことが求められます。この要件は、データ保護を強化するためのものです。この動作によって、データ内で繰り返されるパターンが観測可能になることを防止できます。ただし、JOIN クエリで暗号化されたデータを処理するにあたって、値の重複を気にしない場合は、[複製を許可] パラメータでこの保守的なチェックを無効にできます。

[名前の異なる列の JOIN を許可] パラメータ

コンソールでは、[コラボレーションの作成](#)時に [名前の異なる列の JOIN を許可] パラメータを設定して、異なる名前の列間の JOIN ステートメントをサポートするかどうかを指定できます。

詳細については、[列ヘッダー名の正規化](#)を参照してください。

次の表で、[名前の異なる列の JOIN を許可] パラメータの値について説明します。

パラメータ値	説明
いいえ	名前の異なる fingerprint 列の結合はサポートされません。JOIN ステートメントでは、同じ名前の列でのみ正確な結果が得られます。

パラメータ値	説明
	 Important [いいえ] を指定すると情報セキュリティは強化されますが、列名について事前にコラボレーション参加者の合意が必要になります。fingerprint列として暗号化したときに2つの列の名前が異なり、[名前の異なる列のJOINを許可]が[いいえ]に設定されていると、それらの列のJOINステートメントは結果を生成しません。これは、暗号化後の値が2つの列の間で共有されないためです。
あり	名前の異なるfingerprint列の結合がサポートされます。柔軟性を高めるために、ユーザーはこの値を[はい]に設定できます。これにより、名前に関係なく列に対してJOINステートメントを実行できます。 [はい]に設定すると、C3R暗号化クライアントはfingerprint列を保護する際に列名を考慮しません。その結果、C3Rテーブルの異なるfingerprint列で共通する値が観測可能になります。 例えば、ある行のCity列とState列の両方に暗号化された同一のJOIN値が存在する場合、その値はNew Yorkであると合理的に推測できます。

CreateCollaboration API オペレーションを使用して、dataEncryptionMetadata パラメータの allowJoinsOnColumnsWithDifferentNames 値を true または false に設定できます。API オペレーションの詳細については、「[AWS Clean Rooms API リファレンス](#)」を参照してください。

デフォルトでは、fingerprint 列の暗号化は、「[ステップ 4: 表形式ファイルの暗号化スキーマを生成する](#)」で設定されたその列の targetHeader 設定の影響を受けます。そのため、同じ cleartext 値でも、暗号化対象の fingerprint 列ごとに暗号化表現は異なります。

このパラメータは、cleartext 値が推測されるのを防ぐのに役立つ場合があります。例えば、fingerprint 列 City と State に暗号化された同一の値が表示されている場合、その値は New

York であると合理的に推測できます。ただし、このパラメータを使用するには、クエリで結合されるすべての列が共通の名前になるように、事前に追加の調整が必要になります。

[名前の異なる列の JOIN を許可] パラメータを使用すると、この制限を緩和できます。パラメータ値を Yes に設定すると、名前に関係なく、暗号化されたすべての列を JOIN で一緒に使用できます。

[NULL 値を保存] パラメータ

コンソールでは、[コラボレーションの作成](#)時に [NULL 値を保存] パラメータを設定して、その列に値がないことを示すことができます。

次の表で、[NULL 値を保存] パラメータの値について説明します。

パラメータ値	説明
いいえ	NULL 値は保持されません。NULL 値は暗号化されたテーブルで NULL として表示されません。NULL 値は C3R テーブルに一意のランダム値として表示されます。
あり	NULL 値は保持されます。NULL 値は暗号化されたテーブルで NULL として表示されます。NULL 値の SQL セマンティックが必要な場合は、この値を [はい] に設定できます。その結果、列が暗号化されているかどうか、および [複製を許可] パラメータの設定に関係なく、NULL エントリは C3R テーブルで NULL として表示されます。

CreateCollaboration API オペレーションを使用して、dataEncryptionMetadata パラメータの preserveNulls 値を true または false に設定できます。API オペレーションの詳細については、「[AWS Clean Rooms API リファレンス](#)」を参照してください。

コラボレーションの [NULL 値を保存] パラメータが [いいえ] に設定されている場合の動作は以下の通りです。

1. cleartext 列の NULL エントリは変更されません。
2. 暗号化された fingerprint 列の NULL エントリは、内容を隠すためにランダムな値として暗号化されます。暗号化された列をその cleartext 列の NULL エントリと結合しても、どの NULL エントリとも一致しません。それぞれが一意のランダムコンテンツを受け取るため、一致は生じません。

3. 暗号化されたsealed 列の NULL エントリは暗号化されます。

コラボレーションの [NULL 値を保存] パラメータ値が [はい] に設定されている場合、列が暗号化されているかどうかに関係なく、すべての列の NULL エントリは NULL のままとなります。

[NULL 値を保存] パラメータは、データエンリッチメントなどのシナリオにおいて、情報が欠落し、NULL で表されているエントリを共有する場合に便利です。また、[NULL 値を保存] パラメータは、JOIN または GROUP BY を実行する列に NULL がある場合に、fingerprint または HMAC 形式でも役立ちます。

[複製を許可] パラメータと [NULL 値を保存] パラメータの値が [いいえ] に設定されている場合、1 つの fingerprint 列に複数の NULL エントリがあると、エラーが発生して暗号化が停止します。いずれかのパラメータ値が [はい] に設定されていれば、そのようなエラーは発生しません。

Cryptographic Computing for Clean Rooms のオプションフラグ

以下のセクションでは、表形式のファイルのカスタマイズとテストにおいて、C3R 暗号化クライアントを使って [データを暗号化](#) するときに設定できるオプションフラグについて説明します。

トピック

- [--csvInputNULLValue フラグ](#)
- [--csvOutputNULLValue フラグ](#)
- [--enableStackTraces フラグ](#)
- [--dryRun フラグ](#)
- [--tempDir フラグ](#)

--csvInputNULLValue フラグ

C3R 暗号化クライアントを使用して [データを暗号化](#) するときに、--csvInputNULLValue フラグを使用して入力データの NULL エントリにカスタムエンコーディングを指定できます。

次の表は、このフラグの使用方法とパラメータをまとめたものです。

使用方法	パラメータ
オプション。ユーザーは入力データの NULL エントリにカスタムエンコーディングを指定できます。	入力 CSV ファイルの NULL 値に対するユーザー指定のエンコーディング

NULL エントリとは、特に SQL テーブルのようなよりリッチな表形式のコンテキストでは、欠落したコンテンツとみなされるエントリです。.csv は歴史的な理由からこの特性を明示的にサポートしていませんが、空白だけを含む空のエントリは NULL と見なすのが一般的な慣習です。したがって、これは C3R 暗号化クライアントのデフォルト動作であり、必要に応じてカスタマイズできます。

--csvOutputNULLValue フラグ

C3R 暗号化クライアントを使用して[データを暗号化](#)するときに、--csvOutputNULLValue フラグを使用して出力データの NULL エントリにカスタムエンコーディングを指定できます。

次の表は、このフラグの使用方法とパラメータをまとめたものです。

使用方法	パラメータ
オプション。ユーザーは、NULL エントリの生成された出力ファイルにカスタムエンコーディングを指定できます。	出力 CSV ファイルの NULL 値に対するユーザー指定のエンコーディング

NULL エントリとは、特に SQL テーブルのようなよりリッチな表形式のコンテキストでは、欠落したコンテンツとみなされるエントリです。.csv は歴史的な理由からこの特性を明示的にサポートしていませんが、空白だけを含む空のエントリは NULL と見なすのが一般的な慣習です。したがって、これは C3R 暗号化クライアントのデフォルト動作であり、必要に応じてカスタマイズできます。

--enableStackTraces フラグ

C3R 暗号化クライアントを使用して[データを暗号化](#)するときに --enableStackTraces フラグを使用すると、C3R でエラーが発生したときに、エラー報告用の追加のコンテキスト情報が提供されます。

AWS はエラーを収集しません。エラーが発生した場合は、スタックトレースを使用してエラーを自分でトラブルシューティングするか、スタックトレースを に送信してサポート サポート を依頼してください。

次の表は、このフラグの使用方法とパラメータをまとめたものです。

使用方法	パラメータ
オプション。C3R 暗号化クライアントでエラーが発生したときに、エラー報告用の追加のコンテキスト情報を提供するために使用します。	なし

--dryRun フラグ

C3R 暗号化クライアントの[暗号化](#)コマンドと[復号化](#)コマンドにはオプションの --dryRun フラグがあります。このフラグは、ユーザーが指定した引数をすべて受け取り、その有効性と一貫性をチェックします。

この --dryRun フラグを使用して、スキーマファイルが有効で対応する入力ファイルと一致しているかどうかを確認できます。

次の表は、このフラグの使用方法とパラメータをまとめたものです。

使用方法	パラメータ
オプション。C3R 暗号化クライアントはパラメータの解析とファイルのチェックを行います。が、暗号化や復号化は行いません。	なし

--tempDir フラグ

設定によっては、暗号化されたファイルは暗号化されていないファイルよりもサイズが大きくなるため、一時ディレクトリを使用することができます。また、データセットを正常に機能させるには、コラボレーションごとに暗号化する必要があります。

C3R を使用して[データを暗号化](#)する場合は、--tempDir フラグを使用して、入力の処理中に一時ファイルを作成する場所を指定できます。

次の表は、このフラグの使用方法とパラメータをまとめたものです。

使用方法	パラメータ
ユーザーは、入力の処理中に一時ファイルを作成する場所を指定できます。	デフォルトはシステム一時ディレクトリです。

クエリと Cryptographic Computing for Clean Rooms

このトピックでは、Cryptographic Computing for Clean Rooms で暗号化されたデータテーブルを使用するクエリの作成について情報を提供します。

トピック

- [NULL で分岐するクエリ](#)
- [1 つのソース列を複数のターゲット列にマッピングする](#)
- [JOIN クエリと SELECT クエリの両方に同じデータを使用する](#)

NULL で分岐するクエリ

NULL ステートメントでクエリを分岐するということは、`IF x IS NULL THEN 0 ELSE 1` のような構文を使用することを意味します。

cleartext列では、NULL ステートメントで常にクエリを分岐させることができます。

sealed列やfingerprint列の場合は、[NULL 値を保存] パラメータ (preserveNulls) の値が true に設定されている場合のみ、NULL ステートメントでクエリを分岐させることができます。

これらの制約に違反するクエリは、誤った結果をもたらす可能性があります。

1 つのソース列を複数のターゲット列にマッピングする

1 つのソース列を複数のターゲット列にマッピングできます。例えば、1 つの列で JOIN と SELECT の両方を実行することができます。

詳細については、「[JOIN クエリと SELECT クエリの両方に同じデータを使用する](#)」を参照してください。

JOIN クエリと SELECT クエリの両方に同じデータを使用する

列内のデータが機密情報ではない場合、そのデータはcleartextのターゲット列に表示され、あらゆる目的に使用できます。

列内のデータが機密情報で、JOIN クエリと SELECT クエリの両方に使用する必要がある場合は、そのソース列を出力ファイルの 2 つのターゲット列にマッピングします。1 つの列は type を fingerprint列として暗号化し、もう 1 つの列は type をシール列として暗号化します。C3R 暗号化クライアントのインタラクティブなスキーマ生成では、ヘッダーサフィックスとして `_fingerprint` と `_sealed` が提示されます。これらのヘッダーサフィックスは、このような列をすばやく区別するのに便利な規則です。

C3R 暗号化クライアントのガイドライン

C3R 暗号化クライアントは、組織が機密データをまとめてデータ分析から新しいインサイトを引き出すために使用するツールです。このツールは、任意の当事者およびプロセス AWS で学習できる内容を暗号的に制限します。これはきわめて重要ですが、データを暗号化によって保護するプロセスでは、コンピューティングリソースとストレージリソースの両面で大きなオーバーヘッドが生じる可能性があります。そのため、各設定を使用する際のトレードオフや、必要な暗号化保証を維持しながら最適な設定を行う方法を理解することが重要です。このトピックでは、C3R 暗号化クライアントとスキーマのさまざまな設定がパフォーマンスに与える影響に焦点を当てています。

C3R 暗号化クライアントの暗号化設定はすべて、異なる暗号化保証を提供します。コラボレーションレベルの設定が、デフォルトでは最も安全です。コラボレーションの作成中に追加機能を有効にすると、暗号文で頻度分析などのアクティビティを実行できるようになり、プライバシーの保証が弱まります。これらの設定がどのように使用され、どのような影響を及ぼすかの詳細については、「[the section called “暗号コンピューティング”](#)」を参照してください。

トピック

- [列タイプがパフォーマンスに与える影響](#)
- [暗号文のサイズが予想せず大きくなった場合のトラブルシューティング](#)

列タイプがパフォーマンスに与える影響

C3R ではcleartext、fingerprint、sealedの 3 つの列タイプを使用します。これらの列タイプはそれぞれ異なる暗号化保証を提供し、使用目的も異なります。以下のセクションでは、列タイプがパフォーマンスに与える影響と、各設定がパフォーマンスに与える影響について説明します。

トピック

- [Cleartext列](#)
- [Fingerprint列](#)
- [Sealed列](#)

Cleartext列

Cleartext列は元の形式から変更されておらず、暗号化処理もされていません。この列タイプを設定することはできず、ストレージやコンピューティングのパフォーマンスにも影響はありません。

Fingerprint列

Fingerprint列は複数のテーブルにまたがるデータを結合するために使用されます。そのためには、生成される暗号文のサイズが常に同じでなければなりません。ただし、これらの列はコラボレーションレベルの設定による影響を受けます。Fingerprint列は、入力に含まれるcleartextに応じて、出力ファイルのサイズにさまざまな程度の影響を与える可能性があります。

トピック

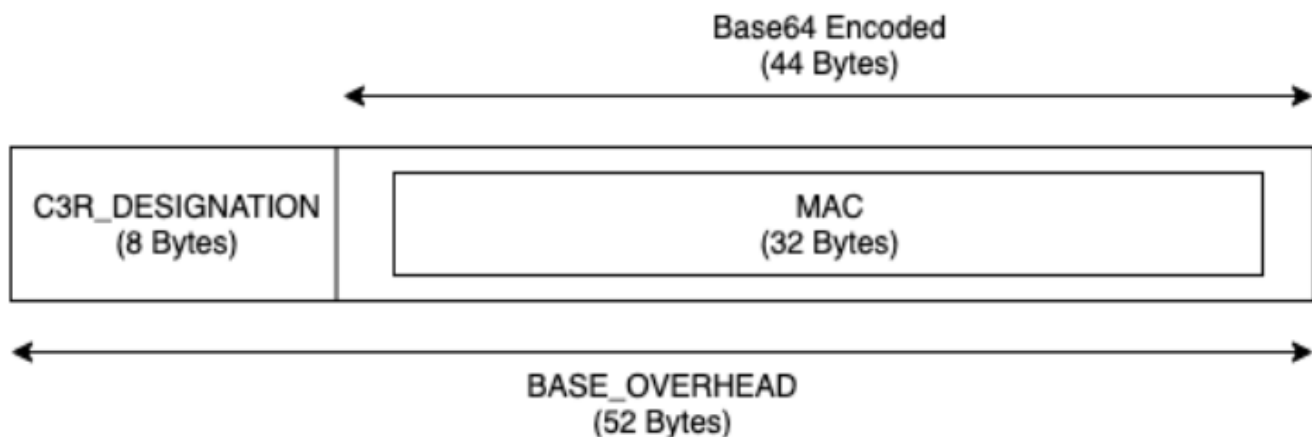
- [fingerprint列の基本オーバーヘッド](#)
- [fingerprint列のコラボレーション設定](#)
- [fingerprint列のデータ例](#)
- [fingerprint列のトラブルシューティング](#)

fingerprint列の基本オーバーヘッド

fingerprint列には基本オーバーヘッドがあります。このオーバーヘッドは一定であり、cleartextのバイトサイズに代わるものです。

fingerprint列内のデータは、Hash-based Message Authentication Code (HMAC)関数によって暗号化処理され、データが 32 バイトのメッセージ認証コード (MAC) に変換されます。その後、このデータは base64 エンコーダで処理され、バイトサイズが約 33% 増加します。先頭には、データが属する列の種類とそれを生成したクライアントバージョンを示す 8 バイトの C3R 指定子が付加されます。最終結果は 52 バイトです。次に、この結果に行数を掛けて、基本オーバーヘッドの合計を求めます (preserveNulls が true に設定されている場合は、null 値以外の合計数を使用します)。

以下の図は、 $BASE\_OVERHEAD = C3R\_DESIGNATION + (MAC * 1.33)$ を表しています。



fingerprint列の出力暗号文は常に 52 バイトです。cleartextの入力データの平均が 52 バイトを超える場合 (完全な住所など) は、これによってストレージサイズが大幅に減少する可能性があります。cleartextの入力データの平均が 52 バイト未満の場合 (顧客の年齢など) は、ストレージサイズが大幅に増える可能性があります。

fingerprint列のコラボレーション設定

preserveNulls の設定

コラボレーションレベルの **preserveNulls** 設定が **false** (デフォルト) の場合、各 **null** 値は固有のランダムな 32 バイトに置き換えられ、**null** ではないかのように処理されます。結果として、各 **null** 値は 52 バイトになります。これにより、データが非常に少ないテーブルでは、この設定が **true** で **null** 値が **null** として渡される場合と比べて、ストレージ要件が大幅に増加する可能性があります。

この設定によるプライバシー保証が不要で、**null** 値をデータセット内に保持する場合は、コラボレーションの作成時に **preserveNulls** 設定を有効にしてください。コラボレーションの作成後に **preserveNulls** 設定を変更することはできません。

fingerprint列のデータ例

以下は、再現可能な設定を含むfingerprint列の入出力データのサンプルセットです。**allowCleartext** や **allowDuplicates** などの他のコラボレーションレベルの設定は結果に影響せず、ローカルで再現を試みる場合は **true** または **false** に設定できます。

共有シークレットの例: wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

コラボレーション ID の例: a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

`allowJoinsOnColumnsWithDifferentNames`: `True`。この設定はパフォーマンスやストレージ要件には影響しません。ただし、この設定を行うと、次の表に示す値を再現するときに、列名の選択は重要ではなくなります。

例 1

Input	null
<code>preserveNulls</code>	TRUE
Output	null
確定的	Yes
入力バイト数	0
出力バイト数	0

例 2

Input	null
<code>preserveNulls</code>	FALSE
Output	01:hmac:31kFjthvV3IUu6mMvFc1a +XAHwgw/Elm0q4p3Yg25kk=
確定的	No
入力バイト数	0
出力バイト数	52

例 3

Input	empty string
<code>preserveNulls</code>	-

Output	01:hmac:oKTgi3Gba+eUb3JteSz 2EMgXUkF1WgM77UP0Ydw5kPQ=
確定的	Yes
入力バイト数	0
出力バイト数	52

例 4

Input	abcdefghijklmnopqrstuvwxy
preserveNulls	-
Output	01:hmac:kU/IqwG7FMmzzshr0B9 scomE0UJUEE7j9keTctp1Gww=
確定的	Yes
入力バイト数	26
出力バイト数	52

例 5

Input	abcdefghijklmnopqrstuvwxyA BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789
preserveNulls	-
Output	01:hmac:ks3htnQbw2vdhCRFF6J NzW5LMndJaHG57uvE26mBtSs=
確定的	Yes
入力バイト数	62

出力バイト数

52

fingerpint列のトラブルシューティング

fingerpint列の暗号文が、入力されたcleartextのサイズより数倍大きいのはなぜですか？

fingerpint列の暗号文の長さは常に 52 バイトです。入力データが小さい場合 (顧客の年齢など) は、サイズが大幅に増加します。この現象は、preserveNulls 設定が false に設定されている場合にも発生する可能性があります。

fingerpint列の暗号文が、入力されたcleartextのサイズより数倍小さいのはなぜですか？

fingerpint列の暗号文の長さは常に 52 バイトです。入力データが大きい場合 (顧客の完全な住所など) は、サイズが大幅に減少します。

preserveNulls による暗号保証が必要かどうかはどうすればわかりますか？

答えは状況により異なります。少なくとも、preserveNulls 設定によってデータがどのように保護されるかを「[the section called “パラメータ”](#)」で確認する必要があります。ただし、組織のデータ処理要件と、それぞれのコラボレーションに適用される契約を参照することをお勧めします。

base64 のオーバーヘッドが発生するのはなぜですか？

CSV などの表形式のファイル形式との互換性を保つには、base64 エンコーディングが必要です。Parquet などの一部のファイル形式はデータのバイナリ表現をサポートしていますが、適切なクエリ結果を得るためには、コラボレーションのすべての参加者が同じ方法でデータを表現することが重要です。

Sealed列

Sealed列は、コラボレーションのメンバー間でデータを転送するために使用します。これらの列の暗号文は非確定的であり、列の構成方法によってはパフォーマンスとストレージの両方に大きな影響を与えます。これらの列は個別に設定でき、多くの場合、C3R 暗号化クライアントのパフォーマンスとそれに伴う出力ファイルサイズに最も大きな影響を与えます。

トピック

- [sealed列の基本オーバーヘッド](#)
- [sealed列のコラボレーション設定](#)
- [スキーマ設定sealed列: パディングタイプ](#)

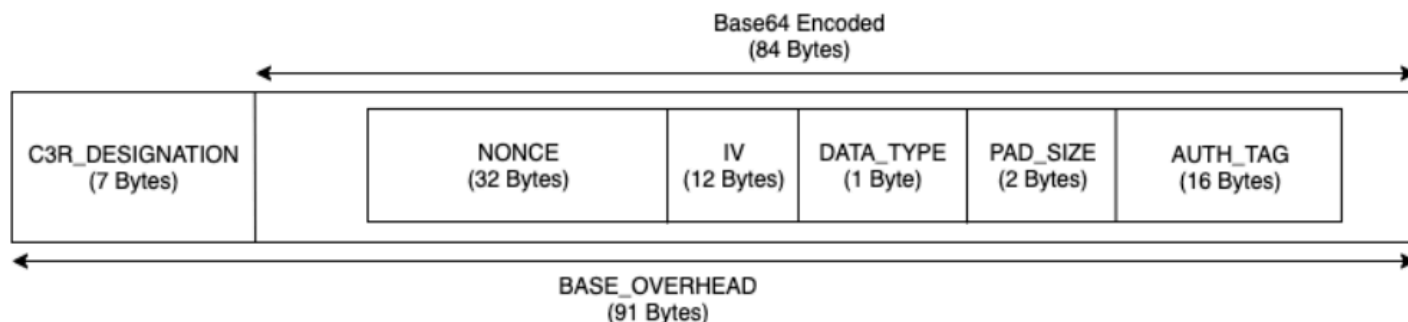
- [sealed列のデータ例](#)
- [sealed列のトラブルシューティング](#)

sealed列の基本オーバーヘッド

sealed列には基本オーバーヘッドがあります。このオーバーヘッドは一定であり、cleartextとパディング (ある場合) のバイトサイズに追加されるものです。

暗号化の前に、sealed列のデータの先頭には、含まれるデータのタイプを示す 1 バイトの文字が付加されます。パディングを選択すると、データがパディングされ、パッドサイズを示す 2 バイトが追加されます。これらのバイトが追加された後、データは AES-GCM を使用して暗号化処理され、IV (12 バイト)、nonce (32 バイト)、Auth Tag (16 バイト) と共に格納されます。その後、このデータは base64 エンコーダで処理され、バイトサイズが約 33% 増加します。データの先頭には、データが属する列の種類とその生成に使用されたクライアントバージョンを示す 7 バイトの C3R 指定子が付加されます。その結果、最終的な基本オーバーヘッドは 91 バイトになります。次に、この結果に行数を掛けて、基本オーバーヘッドの合計を求めます (preserveNulls が true に設定されている場合は、null 値以外の合計数を使用します)。

以下の図は、 $BASE\_OVERHEAD = C3R\_DESIGNATION + ((NONCE + IV + DATA\_TYPE + PAD\_SIZE + AUTH\_TAG) * 1.33)$ を表しています。



sealed列のコラボレーション設定

preserveNulls の設定

コラボレーションレベルの preserveNulls 設定が false (デフォルト) の場合、各 null 値は固有のランダムな 32 バイト値となり、null ではないかのように処理されます。結果として、各 null 値は 91 バイト (パディングされている場合はそれ以上) になります。これにより、データが非常に少ないテーブルでは、この設定が true で null 値が null として渡される場合と比べて、ストレージ要件が大幅に増加する可能性があります。

この設定によるプライバシー保証が不要で、null 値をデータセット内に保持する場合は、コラボレーションの作成時に preserveNulls 設定を有効にしてください。コラボレーションの作成後に preserveNulls 設定を変更することはできません。

スキーマ設定sealed列: パディングタイプ

トピック

- [パディングタイプ none](#)
- [パディングタイプ fixed](#)
- [パディングタイプ max](#)

パディングタイプ none

none のパディングタイプを選択すると、cleartextにパディングは追加されず、前述の基本オーバーヘッドにさらにオーバーヘッドが追加されることもありません。パディングがないと、最もスペース効率の良い出力サイズになります。ただし、fixed や max のパディングタイプと同じプライバシー保証は提供されません。これは、基になるcleartextのサイズが暗号文のサイズから識別できるためです。

パディングタイプ fixed

fixed のパディングタイプを選択すると、列に含まれるデータの長さが隠蔽され、プライバシーを保護する手段となります。これは、暗号化の前に、すべてのcleartextを指定された pad_length にパディングすることで行われます。このサイズを超えるデータがあると、C3R 暗号化クライアントは機能しなくなります。

暗号化の前にcleartextにパディングが追加される場合、AES-GCM で、cleartextと暗号文のバイトとの1対1のマッピングが行われます。base64 エンコーディングによってサイズが 33% 増加します。パディングによって増加するストレージのオーバーヘッドは、pad_length の値からcleartextの長さの平均値を引き、1.33 を掛けることで算出できます。その結果が、レコードごとのパディングの平均オーバーヘッドになります。次に、この結果に行の数を掛けて、パディングのオーバーヘッドの合計を求めます (preserveNulls が true に設定されている場合は、null 値以外の合計数を使用します)。

$$PADDING\_OVERHEAD = (PAD\_LENGTH - AVG\_CLEARTEXT\_LENGTH) * 1.33 * ROW\_COUNT$$

pad_length の最小値には、列内の最大値が収まる値を選択することをお勧めします。例えば、最大値が 50 バイトの場合は、50 バイトの pad_length で十分です。これより大きい値では、ストレージのオーバーヘッドが増えるだけです。

固定長のパディングでは、コンピューティングのオーバーヘッドは大幅に増加しません。

パディングタイプ `max`

`max` のパディングタイプを選択すると、列に含まれるデータの長さが隠蔽され、プライバシーを保護する手段となります。これは、暗号化の前に、すべての cleartext を列内の最大値に追加の `pad_length` を加算した値までパディングすることで行われます。一般に、`max` のパディングは 1 つのデータセットに対して `fixed` のパディングと同様の保証を提供しますが、列内の cleartext の最大値を把握していなくてもかまいません。ただし、更新時には、個々のデータセットの最大値が変わる場合があるため、`max` のパディングで `fixed` のパディングと同様のプライバシー保証が提供されるとは限りません。

`max` のパディングを使用するときは、0 の `pad_length` を追加で選択することをお勧めします。この長さにより、すべての値が列の最大値と同じサイズにパディングされます。これより大きい値では、ストレージのオーバーヘッドが増えるだけです。

特定の列の cleartext の最大値がわかっている場合は、代わりに `fixed` のパディングタイプを使用することをお勧めします。`fixed` のパディングを使用すると、データセットの更新前後で一貫性が保たれます。`max` のパディングを使用すると、データの各サブセットが、そのサブセットにあった最大値までパディングされます。

sealed列のデータ例

以下は、再現可能な設定を含む sealed 列の入出力データのサンプルセットです。`allowCleartext`、`allowJoinsOnColumnsWithDifferentNames`、`allowDuplicates` などの他のコラボレーションレベルの設定は結果に影響せず、ローカルで再現を試みる場合は `true` または `false` に設定できます。これらは再現するための基本設定ですが、sealed 列は非確定的であり、値は毎回変動します。目標は、入力されたバイトと出力されたバイトを比較することです。サンプルの `pad_length` 値は意図的に選択されています。ここでは、`fixed` のパディングが、推奨される最小 `pad_length` 設定を使用した `max` のパディングと同じ値になること、または追加のパディングが望ましいケースが示されています。

共有シークレットの例: `wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

コラボレーション ID の例: `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`

トピック

- [パディングタイプ `none`](#)
- [パディングタイプ `fixed` \(例 1\)](#)
- [パディングタイプ `fixed` \(例 2\)](#)

- [パディングタイプ max \(例 1\)](#)
- [パディングタイプ max \(例 2\)](#)

パディングタイプ **none**

例 1

Input	null
preserveNulls	TRUE
Output	null
確定的	Yes
入力バイト数	0
出力バイト数	0

例 2

Input	null
preserveNulls	FALSE
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfssGSPbNIJfG3iXmu6cbCUrizuV
確定的	No
入力バイト数	0
出力バイト数	91

例 3

Input	empty string

<code>preserveNulls</code>	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstGSPeM6qR8DWC2P B2GM1X41YK
確定的	No
入力バイト数	0
出力バイト数	91

例 4

Input	abcdefghijklmnopqrstuvxyz
<code>preserveNulls</code>	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfsteEE1GKEPiRzyh0 h7t60mWMLTWCv02ckr6pkx9sGL5 VLDQeHzh6DmPpyWNuI=
確定的	No
入力バイト数	26
出力バイト数	127

例 5

Input	abcdefghijklmnopqrstuvxyzA BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789
<code>preserveNulls</code>	-

Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstEE1GKEPiRzyh0 h7t60mWMLTWcV02ckr6plwtH/8t RFnn2rF91bcB9G4+n8GiRfJNmqdP4/ Q0Q3cXb/pbvPcnnohrHIGSX54ua+1/ JfcVjc=
確定的	No
入力バイト数	62
出力バイト数	175

パディングタイプ **fixed** (例 1)

この例では、pad_length は 62 で最大入力値は 62 バイトです。

例 1

Input	null
preserveNulls	TRUE
Output	null
確定的	Yes
入力バイト数	0
出力バイト数	0

例 2

Input	null
preserveNulls	FALSE

Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfssGSNWfMRp7nSb7S MX2s3JKL0hK1+7r75Tk+Mx9jy48 Fcg1y0PvBqRSZ7oqy1V3UKfYTLEZb/ hCz7oaIneVsrcoNpATs0GzbnLkor4L+/ aSuA=
確定的	No
入力バイト数	0
出力バイト数	175

例 3

Input	empty string
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstGSNWfMRp7nSb7S MX2s3JKL0hK1+7r75Tk+Mx9jy48 Fcg1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsrcoLB53l07VZp A60wkuXu29CA=
確定的	No
入力バイト数	0
出力バイト数	175

例 4

Input	abcdefghijklmnopqrstuvwxyz

preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc40TBqfRYZ98t5KU6aWfstEE1GKEPiRzyh0h7t60mWMLTWCv02ckr6pkx9jy48Fcg1y0PvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcutBAc0+Mb9tuU2KIIHH31AWg=
確定的	No
入力バイト数	26
出力バイト数	175

例 5

Input	abcdefghijklmnopqrstuvwxyza BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc40TBqfRYZ98t5KU6aWfstEE1GKEPiRzyh0h7t60mWMLTWCv02ckr6plwtH/8tRFnn2rF91bcB9G4+n8GiRfJNmqdP4/Q0Q3cXb/pbvPcnnohrHIGSX54ua+1/JfcVjc=
確定的	No
入力バイト数	62
出力バイト数	175

パディングタイプ **fixed** (例 2)

この例では、pad_length は 162 で最大入力値は 62 バイトです。

例 1

Input	null
preserveNulls	TRUE
Output	null
確定的	Yes
入力バイト数	0
出力バイト数	0

例 2

Input	null
preserveNulls	FALSE
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmlMDEyMzQ1Njc40TBqfRYZ98t5KU6aWfssGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1y0PvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcnkB0xbLWD7zNdAqQGR0rXoSESdW0I0vpNoGcBfv4cJbG0A3h1DvtkSSVc2B8000GppzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6uwrmwv/xAySX+xcntotL703aBTBb
確定的	No
入力バイト数	0

出力バイト数	307

例 3

Input	empty string
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc4OTBqfRY Z98t5KU6aWfstGSNWfMRp7nSb7S MX2s3JKL0hK1+7r75Tk+Mx9jy48 Fcg1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsrcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwv84lVaT9Yd+6oQx65/+gdVT
確定的	No
入力バイト数	0
出力バイト数	307

例 4

Input	abcdefghijklmnopqrstuvwxyz
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc4OTBqfRY Z98t5KU6aWfsteEE1GKEPiRzyh0 h7t60mWMLTWCv02ckr6pkx9jy48 Fcg1y0PvBqRSZ7oqy1V3UKfYTLE

	Zb/hCz7oaIneVsrcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwT5Hn1+Wyf06ks3QMaRDGSf
確定的	No
入力バイト数	26
出力バイト数	307

例 5

Input	abcdefghijklmnopqrstuvwxyzA BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmlmDEyMzQ1Njc4OTBqfRY Z98t5KU6aWfstEE1GKEPiRzyh0 h7t60mWMLTWCv02ckr6plwtH/8t RFnn2rF91bcB9G4+n8GiRfJNmqd P4/Q0Q3cXb/pbvPcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwjkJXQZ0gPdeFX9Yr/8a1V5i
確定的	No
入力バイト数	62
出力バイト数	307

パディングタイプ **max** (例 1)

この例では、pad_length は 0 で最大入力値は 62 バイトです。

例 1

Input	null
preserveNulls	TRUE
Output	null
確定的	Yes
入力バイト数	0
出力バイト数	0

例 2

Input	null
preserveNulls	FALSE
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfssGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1y0PvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcoNpATs0GzbnLkor4L+/aSuA=
確定的	No
入力バイト数	0
出力バイト数	175

例 3

Input	empty string
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstGSNWfMRp7nSb7S MX2s3JKL0hK1+7r75Tk+Mx9jy48 Fcg1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsircolB53l07VZp A60wkuXu29CA=
確定的	No
入力バイト数	0
出力バイト数	175

例 4

Input	abcdefghijklmnopqrstuvwxyz
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfsteEE1GKEPiRzyh0 h7t60mWMLTWCv02ckr6pkx9jy48 Fcg1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsircutBAc0+Mb9t uU2KIIHH31AWg=
確定的	No
入力バイト数	26
出力バイト数	175

例 5

Input	abcdefghijklmnopqrstuvwxyzA BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstEE1GKEPiRzyh0 h7t60mWMLTWcV02ckr6plwtH/8t RFnn2rF91bcB9G4+n8GiRfJNmqdP4/ Q0Q3cXb/pbvPcnnohrHIGSX54ua+1/ JfcVjc=
確定的	No
入力バイト数	62
出力バイト数	175

パディングタイプ **max** (例 2)

この例では、pad_length は 100 で最大入力値は 62 バイトです。

例 1

Input	null
preserveNulls	TRUE
Output	null
確定的	Yes
入力バイト数	0
出力バイト数	0

例 2

Input	null
preserveNulls	FALSE
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfssGSNWfMRp7nSb7S MX2s3JKL0hK1+7r75Tk+Mx9jy48 Fcgl1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsrcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwv/xAySX+xcntotL703aBTBb
確定的	No
入力バイト数	0
出力バイト数	307

例 3

Input	empty string
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstGSNWfMRp7nSb7S MX2s3JKL0hK1+7r75Tk+Mx9jy48 Fcgl1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsrcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp

	pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwv84lVaT9Yd+6oQx65/+gdVT
確定的	No
入力バイト数	0
出力バイト数	307

例 4

Input	abcdefghijklmnopqrstuvwxyz
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc4OTBqfRY Z98t5KU6aWfstEE1GKEPiRzyh0 h7t60mWMLTWCv02ckr6pkx9jy48 Fcgl1y0PvBqRSZ7oqy1V3UKfYTLE Zb/hCz7oaIneVsrcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwX5Hn1+Wyf06ks3QMaRDGSf
確定的	No
入力バイト数	26
出力バイト数	307

例 5

Input	abcdefghijklmnopqrstuvwxyzA BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789
preserveNulls	-
Output	01:enc:bm9uY2UwMTIzNDU2Nzg5 MG5vbmNlMDEyMzQ1Njc40TBqfRY Z98t5KU6aWfstEE1GKEPiRzyh0 h7t60mWMLTWcV02ckr6plwtH/8t RFnn2rF91bcB9G4+n8GiRfJNmqd P4/Q0Q3cXb/pbvPcnkB0xbLWD7z NdAqQGR0rXoSESdW0I0vpNoGcBf v4cJbG0A3h1DvtkSSVc2B8000Gp pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6 uwrmwjkJXQZ0gPdeFX9Yr/8a1V5i
確定的	No
入力バイト数	62
出力バイト数	307

sealed列のトラブルシューティング

sealed列の暗号文が、入力されたcleartextのサイズより数倍大きいのはなぜですか？

これはいくつかの要因によって異なります。第一に、Cleartext 列の暗号文の長さは必ず 91 バイト以上になります。入力データが小さい場合 (顧客の年齢など) は、サイズが大幅に増加します。第二に、preserveNulls が false に設定されており、入力データに多数の null 値が含まれていた場合、それらの null 値がそれぞれ 91 バイトの暗号文に変換されます。そして最後に、パディングを使用すると、当然のことながら、暗号化の前に cleartext データにバイトが追加されます。

sealed 列のほとんどのデータは非常に小さいのですが、パディングを使う必要があります。スペースを節約するために、大きな値を除外して別途処理することはできますか？

大きな値を除外して別途処理することはお勧めしません。そうすることで、C3R 暗号化クライアントで提供されるプライバシー保証が変わります。脅威の一例として、観察者が暗号化された両方のデータセットを見ることができると仮定します。あるデータサブセットの列のパディングが別のサブセットよりも大幅に大きかったり小さかったりすることを観察者に知られると、各サブセット内のデータのサイズが推測されるおそれがあります。例えば `fullName` 列が、あるファイルでは合計 40 バイトにパディングされ、別のファイルでは 800 バイトにパディングされているとします。この場合観察者は、一方のデータセットに世界で最も長い名前 (747 バイト) が含まれていると仮定する可能性があります。

max のパディングタイプを使用する場合、追加のパディングは必要ですか？

いいえ。**max** のパディングを使用するときは、`pad_length` (列の最大値を超える追加パディングとも呼ばれます) を 0 に設定することをお勧めします。

fixed のパディングを使用するときに、最大値が確実に収まるよう **pad_length** に大きい値を選択してもよいですか？

はい。ただし、パディングの長さが長いと効率が低下し、必要以上に多くのストレージを消費します。最大値の大きさを確認し、`pad_length` にその値を設定することをおすすめします。

preserveNulls による暗号保証が必要かどうかはどうすればわかりますか？

答えは状況により異なります。少なくとも、**preserveNulls** 設定によってデータがどのように保護されるかを「[Cryptographic Computing for Clean Rooms](#)」で確認する必要があります。ただし、組織のデータ処理要件と、それぞれのコラボレーションに適用される契約を参照することをお勧めします。

base64 のオーバーヘッドが発生するのはなぜですか？

CSV などの表形式のファイル形式との互換性を保つには、base64 エンコーディングが必要です。Parquet などの一部のファイル形式はデータのバイナリ表現をサポートしていますが、適切なクエリ結果を得るためには、コラボレーションのすべての参加者が同じ方法でデータを表現することが重要です。

暗号文のサイズが予想せず大きくなった場合のトラブルシューティング

データを暗号化し、生成されたデータのサイズが驚くほど大きくなったとしましょう。次の手順は、サイズが増加している場所と、実行できるアクション (ある場合) を特定するのに役立ちます。

サイズの増加が発生した場所の特定

暗号化されたデータが cleartext データよりも大幅に大きくなった理由をトラブルシューティングする前に、まずサイズが増加している場所を特定する必要があります。Cleartext 列は変更されていないため、無視しても問題ありません。残りの fingerprint 列と sealed 列を見て、大幅に増えている列を 1 つ選びます。

サイズが増加した理由の特定

fingerprint 列または sealed 列がサイズ増加の原因となっている可能性があります。

トピック

- [fingerprint列が原因でサイズの増加が生じている場合](#)
- [sealed 列が原因でサイズの増加が生じている場合](#)

fingerprint列が原因でサイズの増加が生じている場合

ストレージ増加の最も大きな原因となっているのが fingerprint 列である場合は、cleartext データが小さいこと (顧客の年齢など) が要因と考えられます。生成される各 fingerprint 暗号文の長さは 52 バイトになります。残念ながら、この問題について列単位で対処できることは何もありません。ストレージ要件への影響など、この列の詳細については「[fingerprint列の基本オーバーヘッド](#)」を参照してください。

fingerprint 列のサイズが大きくなるもう 1 つの原因として、preserveNulls のコラボレーション設定が考えられます。preserveNulls のコラボレーション設定が無効 (デフォルト設定) になっていると、fingerprint 列の null 値はすべて 52 バイトの暗号文になります。現在のコラボレーションでは、これに対してできることは何もありません。preserveNulls 設定はコラボレーションの作成時に設定され、正しいクエリ結果を得るためにすべてのコラボレーターが同じ設定を使用する必要があります。preserveNulls 設定の詳細と、この設定を有効にした場合にデータのプライバシー保護にどのような影響が及ぶかについては、「[the section called “暗号コンピューティング”](#)」を参照してください。

sealed 列が原因でサイズの増加が生じている場合

ストレージ増加の最も大きな原因となっているのが sealed 列である場合、サイズの増加を引き起こしている可能性のある要因がいくつかあります。

cleartext データが小さい場合 (顧客の年齢など)、生成される各 sealed 暗号文の長さは 91 バイト以上になります。残念ながら、この問題についてできることは何もありません。ストレージ要件への影響など、この列の詳細については「[sealed列の基本オーバーヘッド](#)」を参照してください。

sealed列でストレージが増加する 2 つ目の主な要因はパディングです。パディングとは、データセット内の個々の値のサイズを隠蔽するために、暗号化の前にcleartextに余分なバイトを追加することです。データセットにとって最小限の値でパディングを設定することをお勧めします。少なくとも、fixed パディングの pad_length は、列内で考えられる最大の値が収まるように設定する必要があります。これより大きい値を設定しても、プライバシー保証は追加されません。例えば、列内で考えられる最大の値が 50 バイトであることがわかっている場合は、pad_length を 50 バイトに設定することをお勧めします。ただし、sealed 列で max のパディングを使用している場合は、pad_length を 0 バイトに設定することをお勧めします。これは、max のパディングが、列内の最大値を超える追加のパディングを指すためです。

sealed列のサイズが大きくなる原因として考えられる最後の要因は、preserveNulls のコラボレーション設定です。preserveNulls のコラボレーション設定が無効 (デフォルト設定) になっていると、sealed 列の null 値はすべて 91 バイトの暗号文になります。現在のコラボレーションでは、これに対してできることは何也没有什么ありません。preserveNulls 設定はコラボレーションの作成時に設定され、正しいクエリ結果を得るためにすべてのコラボレーターが同じ設定を使用する必要があります。この設定の詳細と、設定を有効にした場合にデータのプライバシー保護にどのような影響が及ぶかについては、「[the section called “暗号コンピューティング”](#)」を参照してください。

セットアップ AWS Clean Rooms

以下のトピックでは、 を設定する方法について説明します AWS Clean Rooms。

トピック

- [にサインアップする AWS](#)
- [のサービスロールを設定する AWS Clean Rooms](#)
- [AWS Clean Rooms ML のサービスロールを設定する](#)

にサインアップする AWS

AWS Clean Roomsまたは を使用する前に AWS のサービス、AWS で にサインアップする必要があります AWS アカウント。

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/signup> を開きます。
2. オンラインの手順に従います。

サインアップ手順中に、電話のキーパッドに入力する検証コードが記載された電話が届きます。

3. にサインアップすると AWS アカウント、AWS アカウント ルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、[管理ユーザーに管理アクセスを割り当て、ルートユーザーアクセスが必要なタスク](#)を実行する場合にのみ、ルートユーザーを使用してください。

のサービスロールを設定する AWS Clean Rooms

以下のセクションでは、各タスクの実行に必要なロールについて説明します。

トピック

- [管理者ユーザーの作成](#)
- [コラボレーションメンバー用の IAM ロールの作成](#)

- [Amazon S3 からデータを読み取るサービスロールを作成する](#)
- [Amazon Athena からデータを読み取るサービスロールを作成する](#)
- [Snowflake からデータを読み取るサービスロールを作成する](#)
- [S3 バケットからコードを読み取るサービスロールを作成する \(PySpark 分析テンプレートロール\)](#)
- [PySpark ジョブの結果を書き込むサービスロールを作成する](#)
- [結果を受け取るサービスロールを作成する](#)

管理者ユーザーの作成

を使用するには AWS Clean Rooms、管理者ユーザーを作成し、管理者ユーザーを管理者グループに追加する必要があります。

管理者ユーザーを作成するには、以下のいずれかのオプションを選択します。

管理者を管理する方法を 1 つ選択します	目的	方法	以下の操作も可能
IAM アイデンティティセンター内 (推奨)	短期の認証情報を使用して AWS にアクセスします。 これはセキュリティのベストプラクティスと一致しています。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 IAM アイデンティティセンター ユーザーガイドの「 開始方法 」の手順に従います。	AWS Command Line Interface ユーザーガイドの を使用する AWS CLI ようにを設定 AWS IAM アイデンティティセンター して、プログラムによるアクセスを設定します。

管理者を管理する方法を 1 つ選択します	目的	方法	以下の操作も可能
IAM 内 (非推奨)	長期認証情報を使用して AWS にアクセスします。	IAM ユーザーガイドの「 緊急アクセス用の IAM ユーザーを作成する 」の手順に従います。	IAM ユーザーガイドの「 IAM ユーザーのアクセスキーを管理する 」の手順に従って、プログラムによるアクセスを設定します。

コラボレーションメンバー用の IAM ロールの作成

メンバーは、コラボレーションに参加している AWS 顧客です。

コラボレーションメンバー用の IAM ロールを作成するには

- 「AWS Identity and Access Management IAM ユーザーガイド」の「[IAM ユーザーにアクセス許可を委任するロールの作成](#)」を参照してください。
- [ポリシーを作成] ステップで、[ポリシーエディタ] の [JSON] タブを選択し、コラボレーションメンバーに付与された機能に応じてポリシーを追加します。

AWS Clean Rooms では、一般的なユースケースに基づいて以下の管理ポリシーを提供しています。

目的	使用
リソースとメタデータを表示する	AWS 管理ポリシー: AWSCleanRoomsReadOnlyAccess
クエリ	AWS 管理ポリシー: AWSCleanRoomsFullAccess
ジョブのクエリと実行	AWS 管理ポリシー: AWSCleanRoomsFullAccess

目的	使用
クエリを実行して結果を受け取る	AWS 管理ポリシー: AWSCleanRoomsFullAccess
コラボレーションリソースを管理するが、クエリは実行しない	AWS 管理ポリシー: AWSCleanRoomsFullAccessNoQuerying

が提供するさまざまな管理ポリシーの詳細については AWS Clean Rooms、「[IAM ユーザーガイド](#)」の「[AWS Clean Rooms 管理ポリシー](#)」を参照してください。

Amazon S3 からデータを読み取るサービスロールを作成する

AWS Clean Rooms はサービスロールを使用して Amazon S3 からデータを読み込みます。

このサービスロールを作成する方法は 2 つあります。

- サービスロールを作成するために必要な IAM アクセス許可がある場合は、AWS Clean Rooms コンソールを使用してサービスロールを作成します。
- および `iam:CreatePolicy`、`iam:AttachRolePolicy` 許可がない場合 `iam:CreateRole`、または IAM ロールを手動で作成する場合は、次のいずれかを実行します。
 - カスタム信頼ポリシーを使用してサービスロールを作成するには、次の手順に従います。
 - 次の手順を使用してサービスロールを作成するよう、管理者に依頼します。

Note

ユーザーまたは IAM 管理者は、コンソールを使用して AWS Clean Rooms サービスロールを作成するために必要なアクセス許可がない場合にのみ、この手順に従う必要があります。

カスタム信頼ポリシーを使用して Amazon S3 からデータを読み取るサービスロールを作成するには

1. カスタム信頼ポリシーを使用してロールを作成します。詳細については、AWS Identity and Access Management 「[ユーザーガイド](#)」の「[カスタム信頼ポリシー \(コンソール\) を使用したロールの作成](#)」の手順を参照してください。

2. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のカスタム信頼ポリシーを使用します。

 Note

ロールが特定のコラボレーションメンバーシップのコンテキストでのみ使用されるようにしたい場合は、信頼ポリシーをさらに絞り込むことができます。詳細については、「[サービス間の混乱した代理の防止](#)」を参照してください。

JSON

```
{  
  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "RoleTrustPolicyForCleanRoomsService",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "cleanrooms.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

3. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のアクセス許可ポリシーを使用します。

 Note

以下のポリシー例は、AWS Glue メタデータとそれに対応する Amazon S3 データを読み取るのに必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。例えば、Amazon S3 データのカスタム KMS キーを設定している場合は、追加の AWS Key Management Service (AWS KMS) アクセス許可でこのポリシーを修正する必要がある場合があります。

AWS Glue リソースと基盤となる Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じ 必要がある 必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "NecessaryGluePermissions",
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:database/databaseName",
        "arn:aws:glue:us-east-1:111122223333:table/databaseName/tableName",
        "arn:aws:glue:us-east-1:111122223333:catalog"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetSchema",
        "glue:GetSchemaVersion"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "NecessaryS3BucketPermissions",
      "Effect": "Allow",
      "Action": [
```

```

        "s3:GetBucketLocation",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::bucket"
    ],
    "Condition": {
        "StringEquals": {
            "s3:ResourceAccount": [
                "444455556666"
            ]
        }
    }
},
{
    "Sid": "NecessaryS3ObjectPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::bucket/prefix/*"
    ],
    "Condition": {
        "StringEquals": {
            "s3:ResourceAccount": [
                "444455556666"
            ]
        }
    }
}
]
}

```

Note

このポリシーは、データカタログメタデータと実際のデータストレージが異なる当事者によって管理されるコラボレーションをサポートするために AWS Clean Rooms、2 つの AWS アカウント IDs を参照します。

- 111122223333 - AWS Glue Data Catalog リソース (データベース、テーブル、カタログ) を所有するアカウントです。最初のステートメントは、このアカウントの AWS

Glue カタログからテーブルスキーマ、パーティション情報、メタデータにアクセスするアクセス許可を付与します。

- 444455556666 - これは、実際のデータファイルを含む Amazon S3 バケットを所有するアカウントです。Amazon S3 アクセス許可 (ステートメント 3 および 4) は、s3:ResourceAccount条件を通じてこのアカウントが所有するバケットに制限されます。

この設定は、1 つのチームがデータカタログとスキーマ定義を管理し、別のチームが基盤となるデータストレージインフラストラクチャを所有する一般的なエンタープライズデータアーキテクチャをサポートします。このs3:ResourceAccount条件は、Amazon S3 オペレーションが指定されたアカウントが所有するバケットでのみ機能するようにすることで、追加のセキュリティレイヤーを提供します。

4. 各#####を自分の情報に置き換えます。
5. 引き続き、「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従ってロールを作成します。

Amazon Athena からデータを読み取るサービスロールを作成する

AWS Clean Rooms はサービスロールを使用して Amazon Athena からデータを読み込みます。

カスタム信頼ポリシーを使用して Athena からデータを読み取るサービスロールを作成するには

1. カスタム信頼ポリシーを使用してロールを作成します。詳細については、「AWS Identity and Access Management ユーザーガイド」の「[カスタム信頼ポリシー \(コンソール\) を使用したロールの作成](#)」の手順を参照してください。
2. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のカスタム信頼ポリシーを使用します。

Note

ロールが特定のコラボレーションメンバーシップのコンテキストでのみ使用されるようにしたい場合は、信頼ポリシーをさらに絞り込むことができます。詳細については、「[サービス間の混乱した代理の防止](#)」を参照してください。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoleTrustPolicyForCleanRoomsService",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のアクセス許可ポリシーを使用します。

 Note

次のポリシー例では、AWS Glue メタデータとその対応する Athena データを読み取るために必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。例えば、Amazon S3 データのカスタム KMS キーをすでに設定している場合は、追加の AWS KMS アクセス許可でこのポリシーを修正する必要がある場合があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:GetTableMetadata",
        "athena:GetDataCatalog",
        "athena:StartQueryExecution",

```

```

        "athena:GetQueryExecution",
        "athena:GetQueryResults"
    ],
    "Resource": [
        "arn:aws:athena:region:accountId:workgroup/workgroup",
        "arn:aws:athena:region:accountId:datacatalog/federatedCatalogName"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "glue:GetDatabase",
        "glue:GetTable",
        "glue:GetCatalog"
    ],
    "Resource": [
        "arn:aws:glue:region:accountId:catalog",
        "arn:aws:glue:region:accountId:catalog/federatedCatalogName",
        "arn:aws:glue:region:accountId:database/federatedCatalogName/databaseName",
        "arn:aws:glue:region:accountId:table/federatedCatalogName/databaseName/tableName"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:AbortMultipartUpload",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:ListMultipartUploadParts"
    ],
    "Resource": [
        "arn:aws:s3::athenaResultsBucket",
        "arn:aws:s3::athenaResultsBucket/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "accountId"
        }
    }
},

```

```
{
  "Effect": "Allow",
  "Action": "lakeformation:GetDataAccess",
  "Resource": "*"
}
```

4. 各#####を自分の情報に置き換えます。
5. 引き続き、「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従ってロールを作成します。

Lake Formation アクセス許可をセットアップする

Lake Formation アクセス許可で保護されたリソースをクエリする場合、サービスロールには、table/view/catalogに対する Select および Describe アクセス許可と、AWS Glue データベースに対する Describe アクセス許可が必要です。

詳細については、以下を参照してください。

- Amazon Athena [Athena ユーザーガイド](#)の「[Athena を使用して に登録されたデータをクエリ AWS Lake Formation](#)する」
- AWS Lake Formation デベロッパーガイドの [Lake Formation アクセス許可へのオンボーディング](#)

Snowflake からデータを読み取るサービスロールを作成する

AWS Clean Rooms はサービスロールを使用して Snowflake の認証情報を取得し、このソースからデータを読み込みます。

このサービスロールを作成するには、次の 2 つの方法があります。

- サービスロールを作成するために必要な IAM アクセス許可がある場合は、AWS Clean Rooms コンソールを使用してサービスロールを作成します。
- および アクセスiam:CreatePolicyiam:AttachRolePolicy許可がない場合iam:CreateRole、または IAM ロールを手動で作成する場合は、次のいずれかを実行します。
 - カスタム信頼ポリシーを使用してサービスロールを作成するには、次の手順に従います。
 - 次の手順を使用してサービスロールを作成するよう、管理者に依頼します。

Note

ユーザーまたは IAM 管理者は、コンソールを使用して AWS Clean Rooms サービスロールを作成するために必要なアクセス許可がない場合にのみ、この手順に従う必要があります。

カスタム信頼ポリシーを使用して Snowflake からデータを読み取るサービスロールを作成するには

1. カスタム信頼ポリシーを使用してロールを作成します。詳細については、AWS Identity and Access Management 「[ユーザーガイド](#)」の「[カスタム信頼ポリシー \(コンソール\) を使用したロールの作成](#)」の手順を参照してください。
2. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のカスタム信頼ポリシーを使用します。

Note

ロールが特定のコラボレーションメンバーシップのコンテキストでのみ使用されるようにしたい場合は、信頼ポリシーをさらに絞り込むことができます。詳細については、「[サービス間の混乱した代理の防止](#)」を参照してください。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIfSourceArnMatches",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ForAnyValue:ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cleanrooms:us-  
east-1:111122223333:membership/membershipId",

```

```
"arn:aws:cleanrooms:us-east-1:444455556666:membership/queryRunnerMembershipId"
]
}
}
}
```

 Note

この信頼ポリシーは、クエリの実行責任が複数の当事者に分散されるコラボレーションをサポートするために AWS Clean Rooms、2 つの異なる AWS アカウント IDs を参照します。

- 111122223333 - これは、コラボレーションに参加しているメンバーシップを含むアカウントです。このメンバーシップは、ロールアクセスを必要とするデータテーブル、分析ルール、またはその他のコラボレーションリソースを所有している場合があります。
- 444455556666 - これは、クエリの実行を担当するメンバーシップ(「クエリランナー」)を含むアカウントです。このメンバーシップは保護されたクエリを実行し、必要なコンピューティングリソースとデータリソースにアクセスするためにこのロールを引き受ける必要があります。

この設定により、ある当事者がデータまたは分析テンプレートを提供し、別の当事者が実際のクエリを実行するシナリオが可能になります。どちらのロールも、同じ実行ロールを通じて異なるが補完的なアクセス許可を必要とします。このaws:SourceArn条件により、これら 2 つの特定のメンバーシップから発生する AWS Clean Rooms オペレーションのみがロールを引き受けることができるようになり、分散ジョブの実行と結果管理ワークフローをサポートしながらセキュリティが維持されます。

3. カスタム信頼ポリシー (コンソール) を使用してロールを作成する手順に従って、次のいずれかのアクセス許可ポリシーを使用します。

顧客所有の KMS キーで暗号化されたシークレットのアクセス許可ポリシー

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:secretIdentifier",
      "Effect": "Allow"
    },
    {
      "Sid": "AllowDecryptViaSecretsManagerForKey",
      "Action": "kms:Decrypt",
      "Resource": "arn:aws:kms:us-east-1:444455556666:key/keyIdentifier",
      "Effect": "Allow",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "secretsmanager.us-east-1.amazonaws.com",
          "kms:EncryptionContext:SecretARN":
            "arn:aws:secretsmanager:us-east-1:111122223333:secret:secretIdentifier"
        }
      }
    }
  ]
}
```

 Note

このポリシー AWS アカウント IDs を参照します。

- 111122223333 - これはシークレットを所有し、に保存するアカウントです。最初のステートメントは、このアカウントからシークレット値を取得するアクセス許可を付与します。
- 444455556666 - シークレットの暗号化に使用される AWS KMS キーを所有するアカウントです。2 番目のステートメントは、このアカウントの AWS KMS キーを使用してシークレットを復号するアクセス許可を付与します。

この設定は、以下のエンタープライズ環境で一般的です。

- シークレットは 1 つのアカウント (アカウント 1) で一元管理されます。
- 暗号化キーは、別のセキュリティアカウントまたは共有サービスアカウント (アカウント 2) によって管理されます。
- アカウント 2 の AWS KMS キーポリシーでは、アカウント 1 の サービスが暗号化/復号オペレーションに キーを使用することも許可する必要があります。

この `kms:EncryptionContext:SecretARN` 条件により、AWS KMS キーはこの特定のシークレットの復号にのみ使用でき、クロスアカウントアクセスのセキュリティをさらに強化できます。

で暗号化されたシークレットのアクセス許可ポリシー AWS マネージドキー

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:secretIdentifier",
      "Effect": "Allow"
    }
  ]
}
```

4. 各 `#####` を自分の情報に置き換えます。
5. 引き続き、「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従ってロールを作成します。

S3 バケットからコードを読み取るサービスロールを作成する (PySpark 分析テンプレートロール)

AWS Clean Rooms は、PySpark 分析テンプレートを使用するときに、サービスロールを使用してコラボレーションメンバーの指定された S3 バケットからコードを読み込みます。

S3 バケットからコードを読み取るサービスロールを作成するには

1. カスタム信頼ポリシーを使用してロールを作成します。詳細については、「[AWS Identity and Access Management ユーザーガイド](#)」の「[カスタム信頼ポリシー \(コンソール\) を使用したロールの作成](#)」の手順を参照してください。
2. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のカスタム信頼ポリシーを使用します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ForAnyValue:ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cleanrooms:us-east-1:111122223333:membership/jobRunnerMembershipId",
            "arn:aws:cleanrooms:us-east-1:444455556666:membership/analysisTemplateOwnerMembershipId"
          ]
        }
      }
    }
  ]
}
```

Note

この信頼ポリシーは、マルチパーティー AWS Clean Rooms コラボレーションシナリオをサポートするために 2 つの AWS アカウント IDs を参照します。

- 111122223333 - これは、クエリの実行を担当するメンバーシップ (「ジョブランナー」) を含むアカウントです。このメンバーシップは分析ジョブを実行し、必要なリソースにアクセスするためにこのロールを引き受ける必要があります。
- 444455556666 - これは、分析テンプレートとその関連メンバーシップ (「分析テンプレート所有者」) を所有するアカウントです。このメンバーシップは、実行できるクエリを定義し、分析を管理および実行するためにこのロールを引き受ける必要があります。

この設定は、複数の当事者がそれぞれ独自の AWS アカウント とメンバーシップを持つ同じ AWS Clean Rooms コラボレーションに参加するコラボレーションで一般的です。クエリエグゼキューターと分析テンプレート所有者の両方が共有リソースにアクセスする必要があります。このaws:SourceArn条件により、これら 2 つの特定のメンバーシップから発生する AWS Clean Rooms オペレーションのみがロールを引き受けることが保証され、マルチパーティーコラボレーションに正確なアクセスコントロールが提供されます。

3. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のアクセス許可ポリシーを使用します。

Note

次のポリシー例では、Amazon S3 からコードを読み取るために必要なアクセス許可をサポートしています。ただし、S3 データの設定方法によっては、このポリシーを変更する必要があります。

Amazon S3 リソースは、コラボレーション AWS リージョン と同じ AWS Clean Rooms にある必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": ["arn:aws:s3:::s3Path"],
      "Condition": {
        "StringEquals": {
          "s3:ResourceAccount": [
            "s3BucketOwnerAccountId"
          ]
        }
      }
    }
  ]
}
```

4. 各#####はお客様の情報に置き換えてください。

- *s3Path* – コードの S3 バケットの場所。
- *s3BucketOwnerAccountId* – S3 バケット所有者の AWS アカウント ID。
- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *jobRunnerAccountId* – クエリとジョブを実行できるメンバーの AWS アカウント ID。
- *jobRunnerMembershipId* – ジョブをクエリして実行できるメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。これにより、このメンバーがこのコラボレーションで分析を実行する場合にのみ、AWS Clean Rooms がロールを引き受けるようになります。
- *analysisTemplateAccountId* – 分析テンプレートの AWS アカウント ID。
- *analysisTemplateOwnerMembershipId* – 分析テンプレートを所有するメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。

- 引き続き、「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従ってロールを作成します。

PySpark ジョブの結果を書き込むサービスロールを作成する

AWS Clean Rooms はサービスロールを使用して、PySpark ジョブの結果を指定された S3 バケットに書き込みます。

PySpark ジョブの結果を書き込むサービスロールを作成するには

- カスタム信頼ポリシーを使用してロールを作成します。詳細については、AWS Identity and Access Management 「[ユーザーガイド](#)」の「[カスタム信頼ポリシー \(コンソール\) を使用したロールの作成](#)」の手順を参照してください。
- 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のカスタム信頼ポリシーを使用します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ForAnyValue:ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cleanrooms:us-east-1:111122223333:membership/jobRunnerMembershipId",
            "arn:aws:cleanrooms:us-east-1:444455556666:membership/rrMembershipId"
          ]
        }
      }
    }
  ]
}
```

Note

この信頼ポリシーは、AWS Clean Rooms 異なる運用ロールとのコラボレーションをサポートするために、2 つの AWS アカウント IDs を参照します。

- 111122223333 - これは、分析ジョブの実行を担当するメンバーシップ (「ジョブランナー」) を含むアカウントです。このメンバーシップは計算ワークロードを実行し、処理リソースにアクセスするにはこのロールを引き受ける必要があります。
- 444455556666 - これは、結果レシーバー (RR) の責任を持つメンバーシップを含むアカウントです。このメンバーシップは、分析ジョブの出力を受信してアクセスする権限があり、指定された場所に結果を書き込むためのロールアクセスが必要です。

この設定により、一方の当事者が計算分析を実行し、他方の当事者が結果を送受信する AWS Clean Rooms シナリオが可能になります。どちらのロールも、同じ実行ロールを通じて異なるが補完的なアクセス許可を必要とします。この `aws:SourceArn` 条件により、これら 2 つの特定のメンバーシップから発生する AWS Clean Rooms オペレーションのみがロールを引き受け、分散ジョブの実行と結果管理ワークフローをサポートしながらセキュリティを維持できます。

3. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のアクセス許可ポリシーを使用します。

Note

次のポリシー例では、Amazon S3 への書き込みに必要なアクセス許可をサポートしています。ただし、S3 の設定方法によっては、このポリシーの変更が必要になる場合があります。

Amazon S3 リソースは、コラボレーション AWS リージョン と同じ AWS Clean Rooms にある必要があります。

JSON

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::bucket/optionalPrefix/*",
        "Condition": {
          "StringEquals": {
            "s3:ResourceAccount": [
              "s3BucketOwnerAccountId"
            ]
          }
        }
      },
      {
        "Effect": "Allow",
        "Action": [
          "s3:GetBucketLocation",
          "s3:ListBucket"
        ],
        "Resource": "arn:aws:s3:::bucket",
        "Condition": {
          "StringEquals": {
            "s3:ResourceAccount": [
              "s3BucketOwnerAccountId"
            ]
          }
        }
      }
    ]
  }
}

```

4. 各#####を自分の情報に置き換えます。

- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *jobRunnerAccountId* – S3 バケットがある AWS アカウント ID。
- *jobRunnerMembershipId* – ジョブをクエリして実行できるメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。これにより、このメンバーがこのコラボレーションで分析を実行する場合にのみ、AWS Clean Rooms がロールを引き受けるようになります。
- *rrAccountId* – AWS アカウント S3 バケットが配置されている ID。

- ***rrMembershipId*** – 結果を受け取ることができるメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。これにより、このメンバーがこのコラボレーションで分析を実行する場合にのみ、AWS Clean Rooms がロールを引き受けるようになります。
 - ***bucket*** – S3 バケットの名前と場所。
 - ***optionalPrefix*** – 結果を特定の S3 プレフィックスに保存する場合のオプションのプレフィックス。
 - ***s3BucketOwnerAccountId*** – S3 バケット所有者の AWS アカウント ID。
5. 引き続き、「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従ってロールを作成します。

結果を受け取るサービスロールを作成する

Note

結果を受け取るだけのメンバーである (コンソールの [自身のメンバー能力] が [結果を受け取る] だけになっている) 場合は、以下の手順に従ってください。

クエリの実行と結果の受け取りの両方が可能なメンバーである (コンソールの [自身のメンバー能力] が [クエリ] と [結果を受け取る] の両方になっている) 場合、この手順は省略できます。

結果のみを受信できるコラボレーションメンバーの場合、はサービスロール AWS Clean Rooms を使用して、コラボレーション内のクエリされたデータの結果を指定された S3 バケットに書き込みます。

このサービスロールを作成するには、次の 2 つの方法があります。

- サービスロールを作成するために必要な IAM アクセス許可がある場合は、AWS Clean Rooms コンソールを使用してサービスロールを作成します。
- および アクセスiam:CreatePolicyiam:AttachRolePolicy許可がない場合iam:CreateRole、または IAM ロールを手動で作成する場合は、次のいずれかを実行します。
 - カスタム信頼ポリシーを使用してサービスロールを作成するには、次の手順に従います。
 - 次の手順を使用してサービスロールを作成するよう、管理者に依頼します。

Note

ユーザーまたは IAM 管理者は、コンソールを使用して AWS Clean Rooms サービスロールを作成するために必要なアクセス許可がない場合にのみ、この手順に従う必要があります。

カスタム信頼ポリシーを使用して結果を受信するサービスロールを作成するには

1. カスタム信頼ポリシーを使用してロールを作成します。詳細については、AWS Identity and Access Management 「[ユーザーガイド](#)」の「[カスタム信頼ポリシー \(コンソール\) を使用したロールの作成](#)」の手順を参照してください。
2. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のカスタム信頼ポリシーを使用します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIfExternalIdMatches",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "sts:ExternalId":
            "arn:aws*:region*:dbuser:*/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa*"
        }
      }
    },
    {
      "Sid": "AllowIfSourceArnMatches",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
```

```

        "ForAnyValue:ArnEquals": {
            "aws:SourceArn": [
                "arn:aws:cleanrooms:us-
east-1:555555555555:membership/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
            ]
        }
    }
}

```

3. 「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従って、次のアクセス許可ポリシーを使用します。

Note

以下のポリシー例は、AWS Glue メタデータとそれに対応する Amazon S3 データを読み取るのに必要なアクセス許可をサポートしています。ただし、S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。

AWS Glue リソースと基盤となる Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じ 必要があります。

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket_name"
            ],
            "Condition": {
                "StringEquals": {

```

```

        "aws:ResourceAccount": "accountId"
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name/optional_key_prefix/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "accountId"
        }
      }
    }
  ]
}

```

4. 各#####を自分の情報に置き換えます。

- **region** – AWS リージョンの名前。例えば、**us-east-1**。
- **a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa** – クエリが行えるメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。これにより、このメンバーがこのコラボレーションで分析を実行する場合にのみ、AWS Clean Rooms がロールを引き受けるようになります。
- **arn:aws:cleanrooms:us-east-1:555555555555:membership/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa** – クエリを実行できるメンバーの単一のメンバーシップ ARN。[メンバーシップ ARN] はコラボレーションの [詳細] タブにあります。これにより、AWS Clean Rooms は、このメンバーがこのコラボレーションで分析を実行する場合にのみロールを引き受けます。
- **bucket_name** – S3 バケットの Amazon リソースネーム (ARN)。[Amazon リソースネーム (ARN)] は Amazon S3 のバケットの [プロパティ] タブにあります。
- **accountId** – S3 バケットが配置されている AWS アカウント ID。

bucket_name/optional_key_prefix – Amazon S3 の結果送信先の Amazon リソースネーム (ARN)。Amazon S3 [Amazon リソースネーム (ARN)] は Amazon S3 のバケットの [プロパティ] タブにあります。

- 引き続き、「[カスタム信頼ポリシーを使用したロールの作成 \(コンソール\)](#)」の手順に従ってロールを作成します。

AWS Clean Rooms ML のサービスロールを設定する

類似モデリングの実行に必要なロールは、カスタムモデルを使用するために必要なロールとは異なります。以下のセクションでは、各タスクの実行に必要なロールについて説明します。

トピック

- [類似モデリングのサービスロールを設定する](#)
- [カスタムモデリングのサービスロールを設定する](#)

類似モデリングのサービスロールを設定する

トピック

- [トレーニングデータを読み取るサービスロールの作成](#)
- [サービスロールを作成して類似セグメントを書き込む](#)
- [シードデータを読み取るサービスロールの作成](#)

トレーニングデータを読み取るサービスロールの作成

AWS Clean Rooms は、サービスロールを使用してトレーニングデータを読み取ります。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

データセットをトレーニングするサービスロールの作成

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のポリシーをコピーして貼り付けます。

Note

次のポリシー例では、AWS Glue メタデータとその対応する Amazon S3 データを読み取るために必要なアクセス許可をサポートしています。ただし、S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

AWS Glue リソースと基盤となる Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じ 必要がある 必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetPartitions",
        "glue:GetPartition",
        "glue:BatchGetPartition",
        "glue:GetUserDefinedFunctions"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:database/databases",
        "arn:aws:glue:us-east-1:111122223333:table/databases/tables",
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/default"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase"
      ],
      "Resource": [
```

```

        "arn:aws:glue:us-east-1:111122223333:database/default"
    ],
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Resource": [
            "arn:aws:s3:::bucket"
        ],
        "Condition": {
            "StringEquals": {
                "s3:ResourceAccount": [
                    "111122223333"
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::bucketFolders/*"
        ],
        "Condition": {
            "StringEquals": {
                "s3:ResourceAccount": [
                    "111122223333"
                ]
            }
        }
    }
]
}

```

KMS キーを使用してデータを復号する必要がある場合、前のテンプレートに AWS KMS ステートメントを追加します。

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
    ],
    "Resource": [
        "arn:aws:kms:region:accountId:key/keyId"
    ],
    "Condition": {
        "ArnLike": {
            "kms:EncryptionContext:aws:s3:arn":
                "arn:aws:s3:::bucketFolders*"
        }
    }
}
```

5. 各#####を自分の情報に置き換えます。

- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *accountId* – S3 バケットがある AWS アカウント ID。
- *database/databases*、*table/databases/tables*、*Catalog*、および *database/default* – アクセス AWS Clean Rooms する必要があるトレーニングデータの場所。
- *bucket* – S3 バケットの Amazon リソースネーム (ARN)。[Amazon リソースネーム (ARN)] は Amazon S3 のバケットの [プロパティ] タブにあります。
- *bucketFolders* – アクセス AWS Clean Rooms する必要がある S3 バケット内の特定のフォルダの名前。

6. [次へ] を選択します。

7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。

8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。
11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。
12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-ml.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEqualsIfExists": {
          "aws:SourceAccount": ["111122223333"]
        },
        "ArnLikeIfExists": {
          "aws:SourceArn": "arn:aws:cleanrooms-ml:us-east-1:111122223333:training-dataset/*"
        }
      }
    }
  ]
}
```

SourceAccount は常にお客様のもので AWS アカウント。SourceArn は特定のトレーニングデータセットに制限できますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

accountId は、トレーニングデータを含む AWS アカウント の ID です。

13. [次へ] を選択し、[アクセス許可を追加] で、作成したポリシーの名前を入力します。(ページを再度読み込む必要がある場合があります)。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。

15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された passRole アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。
- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

サービスロールを作成して類似セグメントを書き込む

AWS Clean Rooms はサービスロールを使用して類似セグメントをバケットに書き込みます。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

サービスロールを作成して類似セグメントを書き込むには

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のポリシーをコピーして貼り付けます。

 Note

次のポリシー例では、AWS Glue メタデータとその対応する Amazon S3 データを読み取るために必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

AWS Glue リソースと基盤となる Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じ 必要がある があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::buckets"
      ],
      "Condition": {
        "StringEquals": {
          "s3:ResourceAccount": [
            "accountId"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketFolders/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:ResourceAccount": [
            "accountId"
          ]
        }
      }
    }
  ]
}
```

```
    ]
  }
}
```

KMS キーを使用してデータを暗号化する必要がある場合、テンプレートに、この AWS KMS ステートメントを追加します。

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:ReEncrypt*",
    ],
    "Resource": [
        "arn:aws:kms:region:accountId:key/keyId"
    ],
    "Condition": {
        "ArnLike": {
            "kms:EncryptionContext:aws:s3:arn":
                "arn:aws:s3:::bucketFolders*"
        }
    }
}
```

5. 各#####はお客様の情報に置き換えてください。

- *buckets* – S3 バケットの Amazon リソースネーム (ARN)。[Amazon リソースネーム (ARN)] は Amazon S3 のバケットの [プロパティ] タブにあります。
- *accountId* – S3 バケットがある AWS アカウント ID。
- *bucketFolders* – アクセス AWS Clean Rooms する必要がある S3 バケット内の特定のフォルダの名前。
- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *keyId* – データの暗号化に必要な KMS キー。

6. [次へ] を選択します。

7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。

8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。
11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。
12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-m1.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEqualsIfExists": {
          "aws:SourceAccount": ["111122223333"]
        },
        "ArnLikeIfExists": {
          "aws:SourceArn": "arn:aws:cleanrooms-m1:us-east-1:111122223333:configured-audience-model/*"
        }
      }
    }
  ]
}
```

SourceAccount は常にお客様のもので AWS アカウント。SourceArn は特定のトレーニングデータセットに制限できますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

13. [次へ] を選択します。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。
15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された passRole アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。
- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

シードデータを読み取るサービスロールの作成

AWS Clean Rooms はサービスロールを使用してシードデータを読み取ります。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

S3 バケットに保存されているシードデータを読み取るサービスロールを作成するには。

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のいずれかのポリシーをコピーして貼り付けます。

Note

次のポリシー例では、AWS Glue メタデータとその対応する Amazon S3 データを読み取るために必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

AWS Glue リソースと基盤となる Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じ 必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::buckets"
      ],
      "Condition": {
        "StringEquals": {
          "s3:ResourceAccount": [
            "accountId"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketFolders/*"
      ],
      "Condition": {
```

```

        "StringEquals": {
            "s3:ResourceAccount": [
                "accountId"
            ]
        }
    }
}
]
}

```

Note

次のポリシー例は、SQL クエリの結果を読み取って入力データとして使用するために必要なアクセス許可をサポートしています。ただし、クエリの構造によっては、このポリシーの変更が必要になる場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCleanRoomsStartQuery",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:GetCollaborationAnalysisTemplate",
        "cleanrooms:GetSchema",
        "cleanrooms:StartProtectedQuery"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowCleanRoomsGetAndUpdateQuery",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:GetProtectedQuery",
        "cleanrooms:UpdateProtectedQuery"
      ],
    }
  ]
}

```



```

        "Resource": [
            "arn:aws:cleanrooms:us-east-1:111122223333:membership/queryRunnerMembershipId"
        ]
    }
]
}

```

KMS キーを使用してデータを復号化する必要がある場合、テンプレートに AWS KMS ステートメントを追加します。

```

{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
    ],
    "Resource": [
        "arn:aws:kms:region:accountId:key/keyId"
    ],
    "Condition": {
        "ArnLike": {
            "kms:EncryptionContext:aws:s3:arn":
                "arn:aws:s3:::bucketFolders*"
        }
    }
}

```

5. 各#####はお客様の情報に置き換えてください。

- *buckets* – S3 バケットの Amazon リソースネーム (ARN)。[Amazon リソースネーム (ARN)] は Amazon S3 のバケットの [プロパティ] タブにあります。
- *accountId* – S3 バケットがある AWS アカウント ID。
- *bucketFolders* – アクセス AWS Clean Rooms する必要がある S3 バケット内の特定のフォルダの名前。
- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *queryRunnerAccountId* – クエリを実行するアカウントの AWS アカウント ID。

- **queryRunnerMembershipId** – クエリできるメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。これにより、このメンバーがこのコラボレーションで分析を実行する場合にのみ、AWS Clean Rooms がロールを引き受けるようになります。
- **keyId** – データの暗号化に必要な KMS キー。

6. [次へ] を選択します。
7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。
8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。
11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。
12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-ml.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEqualsIfExists": {
          "aws:SourceAccount": ["111122223333"]
        },
        "ArnLikeIfExists": {
```

```
"aws:SourceArn": "arn:aws:cleanrooms-ml:us-east-1:111122223333:audience-generation-job/*"
    }
  }
}
```

SourceAccount は常にお客様のもので AWS アカウント。SourceArn は特定のトレーニングデータセットに制限できますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

13. [次へ] を選択します。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。
15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された passRole アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。
- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

カスタムモデリングのサービスロールを設定する

トピック

- [カスタム ML モデリングのサービスロールを作成する - ML 設定](#)
- [カスタム ML モデルを提供するサービスロールを作成する](#)
- [データセットをクエリするサービスロールを作成する](#)

- サービスロールを作成して、設定済みテーブルの関連付けを作成する

カスタム ML モデリングのサービスロールを作成する - ML 設定

AWS Clean Rooms はサービスロールを使用して、カスタム ML 設定を作成できるユーザーを制御します。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

このロールでは、[PutMLConfiguration](#) アクションを使用できます。

カスタム ML 設定の作成を許可するサービスロールを作成するには

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のポリシーをコピーして貼り付けます。

Note

次のポリシー例では、S3 バケットへのデータへのアクセスと書き込み、および CloudWatch メトリクスの発行に必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じにある必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowS3ObjectWriteForExport",
      "Effect": "Allow",
```

```

        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::bucket/*"
        ],
        "Condition": {
            "StringEquals": {
                "s3:ResourceAccount": [
                    "111122223333"
                ]
            }
        }
    },
    {
        "Sid": "AllowS3KMSEncryptForExport",
        "Effect": "Allow",
        "Action": [
            "kms:Encrypt",
            "kms:GenerateDataKey*"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:111122223333:key/keyId"
        ],
        "Condition": {
            "StringLike": {
                "kms:EncryptionContext:aws:s3:arn":
                    "arn:aws:s3:::bucket*"
            }
        }
    },
    {
        "Sid": "AllowCloudWatchMetricsPublishingForTrainingJobs",
        "Action": "cloudwatch:PutMetricData",
        "Resource": "*",
        "Effect": "Allow",
        "Condition": {
            "StringLike": {
                "cloudwatch:namespace": "/aws/cleanroomsm1/*"
            }
        }
    },
    {
        "Sid": "AllowCloudWatchLogsPublishingForTrainingOrInferenceJobs",

```

```

        "Effect": "Allow",
        "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:us-east-1:111122223333:log-group:/aws/cleanroomsm1/*"
        ]
    }
}

```

5. 各#####はお客様の情報に置き換えてください。

- bucket – S3 バケットの Amazon リソースネーム (ARN)。[Amazon リソースネーム (ARN)] は Amazon S3 のバケットの [プロパティ] タブにあります。
- **region** – AWS リージョンの名前。例えば、**us-east-1**。
- **accountId** – S3 バケットがある AWS アカウント ID。
- **keyId** – データの暗号化に必要な KMS キー。

6. [次へ] を選択します。

7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。

8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。

11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。

12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-m1.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cleanrooms:us-east-1:111122223333:membership/membershipID"
        }
      }
    }
  ]
}
```

SourceAccount は常にお客様のもので AWS アカウント。SourceArn は特定のトレーニングデータセットに制限できますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

13. [次へ] を選択します。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。
15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された passRole アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。

- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

カスタム ML モデルを提供するサービスロールを作成する

AWS Clean Rooms はサービスロールを使用して、カスタム ML モデルアルゴリズムを作成できるユーザーを制御します。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

このロールでは、[CreateConfiguredModelAlgorithm](#) アクションを使用できます。

メンバーがカスタム ML モデルを提供できるようにするサービスロールを作成するには

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のポリシーをコピーして貼り付けます。

Note

次のポリシー例では、モデルアルゴリズムを含む docker イメージを取得するために必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じにある必要があります。

JSON

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



```

    "Statement": [
      {
        "Sid": "AllowECRIImageDownloadForTrainingAndInferenceJobs",
        "Effect": "Allow",
        "Action": [
          "ecr:BatchGetImage",
          "ecr:BatchCheckLayerAvailability",
          "ecr:GetDownloadUrlForLayer"
        ],
        "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/repoName"
      }
    ]
  }
}

```

5. 各#####を自分の情報に置き換えます。

- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *accountId* – S3 バケットがある AWS アカウント ID。
- *repoName* – データを含むリポジトリの名前。

6. [次へ] を選択します。

7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。

8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。

11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。

12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```

{
  "Version": "2012-10-17",

```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Principal": {  
      "Service": "cleanrooms-ml.amazonaws.com"  
    },  
    "Action": "sts:AssumeRole"  
  }  
]  
}
```

SourceAccount は常に です AWS アカウント 。 は、特定のトレーニングデータセットに制限SourceArnできますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

13. [次へ] を選択します。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。
15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された passRole アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。
- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

データセットをクエリするサービスロールを作成する

AWS Clean Rooms は、サービスロールを使用して、カスタム ML モデリングに使用されるデータセットをクエリできるユーザーを制御します。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

このロールでは、[CreateMLInputChannel](#) アクションを使用できます。

メンバーがデータセットをクエリできるようにするサービスロールを作成するには

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のポリシーをコピーして貼り付けます。

 Note

次のポリシー例では、カスタム ML モデリングに使用されるデータセットのクエリに必要なアクセス許可をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョン と同じにある必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCleanRoomsStartQueryForMLInputChannel",
      "Effect": "Allow",
      "Action": "cleanrooms:StartProtectedQuery",
      "Resource": "*"
    },
    {
      "Sid":
"AllowCleanroomsGetSchemaAndGetAnalysisTemplateForMLInputChannel",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:GetSchema",
        "cleanrooms:GetCollaborationAnalysisTemplate"
      ]
    }
  ]
}
```

```

        "Resource": "*"
      },
      {
        "Sid": "AllowCleanRoomsGetAndUpdateQueryForMLInputChannel",
        "Effect": "Allow",
        "Action": [
          "cleanrooms:GetProtectedQuery",
          "cleanrooms:UpdateProtectedQuery"
        ],
        "Resource": [
          "arn:aws:cleanrooms:us-east-1:111122223333:membership/queryRunnerMembershipId"
        ]
      }
    ]
  }
}

```

5. 各#####を自分の情報に置き換えます。

- *region* – AWS リージョンの名前。例えば、**us-east-1**。
- *queryRunnerAccountId* – クエリを実行するアカウントの AWS アカウント ID。
- *queryRunnerMembershipId* – クエリできるメンバーのメンバーシップ ID。[メンバーシップ ID] はコラボレーションの [詳細] タブにあります。これにより、このメンバーがこのコラボレーションで分析を実行する場合にのみ、AWS Clean Rooms がロールを引き受けるようになります。

6. [次へ] を選択します。

7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。

8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。

11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。

12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-ml.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

SourceAccount は常に です AWS アカウント 。 は、特定のトレーニングデータセットに制限SourceArnできますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

13. [次へ] を選択します。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。
15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された passRole アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。
- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

サービスロールを作成して、設定済みテーブルの関連付けを作成する

AWS Clean Rooms はサービスロールを使用して、設定済みテーブルの関連付けを作成できるユーザーを制御します。必要な IAM アクセス許可がある場合には、コンソールを使用してこのロールを作成できます。CreateRole アクセス許可がない場合は、管理者にサービスロールの作成を依頼してください。

このロールでは、CreateConfiguredTableAssociation アクションを使用できます。

設定されたテーブルの関連付けの作成を許可するサービスロールを作成するには

1. 管理者アカウントを使用して、IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
2. [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーエディタ] で [JSON] タブを選択し、次のポリシーをコピーして貼り付けます。

Note

次のポリシー例では、設定済みテーブルの関連付けの作成をサポートしています。ただし、Amazon S3 データの設定方法によっては、このポリシーを変更する必要がある場合があります。このポリシーには、データを復号するための KMS キーは含まれていません。

Amazon S3 リソースは、AWS Clean Rooms コラボレーション AWS リージョンと同じにある必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/KMS-key-ID",
      "Effect": "Allow"
    }
  ]
}
```

```

    },
    {
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::bucket-name",
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetPartitions",
        "glue:GetPartition",
        "glue:BatchGetPartition"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/Glue database name",
        "arn:aws:glue:us-east-1:111122223333:table/Glue database name/Glue table name"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "glue:GetSchema",
        "glue:GetSchemaVersion"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}

```

❶ プレースホルダーリソース ARNs

このポリシーを使用する場合は、プレースホルダーリソース識別子をリソースの実際の ARNs に置き換える必要があります。

- AWS KMS キーリソース: **KMS-key-ID** を Amazon S3 データを暗号化する実際の AWS KMS キー ID に置き換えます。キーは、AWS Glue カタログリソースを所有するのと同じアカウント (111122223333) にある必要があります。
- Amazon S3 バケットリソース: **bucket-name** を、AWS Glue テーブルデータを含む Amazon S3 バケットの実際の名前に置き換えます。バケット名はグローバルに一意であるため、Amazon S3 バケット ARNs にはアカウント IDs が含まれないことに注意してください。
- AWS Glue リソース: 次のプレースホルダーを実際のリソース名に置き換えます。
 - **Glue #####** - AWS Glue データベースの名前
 - **Glue #####** - AWS Glue テーブルの名前

一貫したアクセス許可を確保するには、すべての AWS Glue リソース (カタログ、データベース、テーブル) が同じ AWS アカウント (111122223333) にある必要があります。このアカウントは、データ暗号化に使用される AWS KMS キーを所有するアカウントと同じにし、AWS Clean Rooms データリソースの統合セキュリティ境界を作成する必要があります。

5. 各#####はお客様の情報に置き換えてください。

- **Amazon S3 ##### KMS ##** – Amazon S3 データの暗号化に使用された KMS キー。データを復号するには、データの暗号化に使用したのと同じ KMS キーを指定する必要があります。
- **AWS Glue ##### Amazon S3 #####** – データを含む AWS Glue テーブルを含む Amazon S3 バケットの名前。
- **region** – AWS リージョンの名前。例えば、**us-east-1**。
- **accountId** – データを所有するアカウントの AWS アカウント ID。
- **AWS Glue #####** – データを含む AWS Glue データベースの名前。
- **AWS Glue #####** – データを含む AWS Glue テーブルの名前。

6. [次へ] を選択します。

7. [確認して作成] で [ポリシー名] と [説明] を入力し、[概要] を確認します。
8. [Create policy] (ポリシーの作成) を選択します。

のポリシーを作成しました AWS Clean Rooms。

9. [アクセス管理] で、[ロール] を選択します。

[ロール] を使用すると、短期間の認証情報を作成できるため、セキュリティ強化のためにお勧めです。[ユーザー] を選択して長期間の認証情報を作成することもできます。

10. [ロールの作成] を選択してください。
11. [ロールの作成] ウィザードの [信頼されたエンティティタイプ] で [カスタム信頼ポリシー] を選択します。
12. 次のカスタム信頼ポリシーをコピーして JSON エディタに貼り付けます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-m1.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

SourceAccount は常に です AWS アカウント 。 は、特定のトレーニングデータセットに制限SourceArnできますが、そのデータセットが作成された後に限られます。トレーニングデータセット ARN がまだわからないため、ワイルドカードはここで指定します。

13. [次へ] を選択します。
14. 作成したポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。
15. [名前、確認、および作成] で、[ロール名] と [説明] を入力します。

 Note

[ロール名] は、クエリを実行して結果を受け取ることができるメンバーとメンバーロールに付与された `passRole` アクセス許可のパターンと一致している必要があります。

- a. [信頼されたエンティティを選択] を確認し、必要に応じて編集します。
- b. [許可を追加] でアクセス許可を確認し、必要に応じて編集します。
- c. [タグ] を確認し、必要に応じてタグを追加します。
- d. [ロールの作成] を選択してください。

のサービスロールを作成しました AWS Clean Rooms。

でのコラボレーションとメンバーシップ AWS Clean Rooms

コラボレーションは、メンバーが設定されたテーブルで分析を実行できる安全な論理境界 AWS Clean Rooms です。

のどのメンバーでもコラボレーション AWS Clean Rooms を作成できます。

コラボレーションクリエイターは、設定済みテーブルを分析して結果を受け取る単一のメンバーを指定できます。ただし、コラボレーションクリエイターは、分析を実行できるメンバーがクエリ結果にアクセスできないようにしたい場合があります。この場合、コラボレーションクリエイターは、[クエリを実行できるメンバーを 1 人](#)、[クエリとジョブを実行できるメンバーを 1 人](#)、[結果を受信できる別のメンバー](#)に指定できます。

ほとんどの場合、クエリを実行できるメンバー、またはジョブをクエリして実行できるメンバーも、[コンピューティングコストを支払うメンバー](#)です。ただし、コラボレーションクリエイターは、別のメンバーがクエリの計算コストを負担するよう設定することもできます。

AWS SDKs を使用してコラボレーションを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

トピック

- [コラボレーションの作成](#)
- [メンバーシップの作成とコラボレーションへの参加](#)
- [コラボレーションの編集](#)
- [の変更リクエストAWS Clean Rooms](#)
- [コラボレーションの削除](#)
- [コラボレーションの表示](#)
- [コラボレーションへのメンバーの招待](#)
- [メンバーのモニタリング](#)
- [コラボレーションへのメンバーの追加](#)
- [コラボレーションからメンバーを削除する](#)
- [コラボレーションからの退出](#)

コラボレーションの作成

コラボレーションを作成する方法は 3 つあります AWS Clean Rooms。

最も基本的な形式は、[クエリのコラボレーション](#)です。このコラボレーションでは、SQL クエリ分析に焦点を当て、クエリを実行できるメンバーと結果を受信できるメンバーの 2 つの主要なロールを持つシンプルな構造を維持します。この基本的なコラボレーション設定は、シンプルなデータ分析タスクに適しています。

2 番目の形式である [クエリとジョブのコラボレーション](#) は、SQL クエリと PySpark ジョブの両方を組み込むことで機能を拡張します。このコラボレーション設定は、同じ基本的なロール構造を維持しますが、アクセス許可を拡張してジョブ実行を含めます。注目すべき要件は、PySpark 分析テンプレートを作成するメンバーも結果を受け取るメンバーでなければならない、分析プロセスにおける明確な説明責任を確保することです。

3 番目の形式である [ML モデリングのコラボレーション](#) は、機械学習ワークフロー用に構築されています。このコラボレーション設定では、2 つのロールが追加されます。1 つはトレーニング済みモデルの結果を必要とするユーザー用、もう 1 つは予測を行うためにこれらのモデルを使用した結果を必要とするユーザー用です。このコラボレーション設定は、コラボレーションメンバーが複雑なデータプロジェクトで連携し、全員のロールとアクセス許可を明確にするのに役立ちます。

以下のトピックでは、クエリ、ジョブ、ML モデリングのコラボレーションを作成する方法について説明します。

トピック

- [クエリのコラボレーションの作成](#)
- [クエリとジョブのコラボレーションの作成](#)
- [ML モデリング用のコラボレーションの作成](#)

クエリのコラボレーションの作成

この手順では、[コラボレーション作成者](#)として次のタスクを実行します。

- [コラボレーションを作成します。](#)
- 1 人以上の [メンバー](#) を [コラボレーション](#) に招待します。
- [クエリを実行できるメンバー](#) や [結果を受け取ることができるメンバー](#) など、メンバーに機能を割り当てる。

コラボレーション作成者が結果を受け取ることができるメンバーでもある場合は、結果の送信先と形式を指定します。また、結果の送信先に結果を書き込むためのサービスロール Amazon リソースネーム (ARN) も提供します。

- [コラボレーションにおけるコンピューティングコストの支払いを担当するメンバー](#)を設定します。

開始する前に、以下の前提条件を満たしていることを確認してください。

- コラボレーションに招待する各メンバーの名前と AWS アカウント ID があります。
- 各メンバーの名前と AWS アカウント ID をコラボレーションのすべてのメンバーと共有するためのアクセス許可があります。

 Note

コラボレーションの作成後にメンバーを追加することはできません。

AWS SDKs を使用してコラボレーションを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

クエリのコラボレーションを作成するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエーターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 右上隅にある [コラボレーションを作成] を選択します。
4. [ステップ 1: コラボレーションの定義] で以下の操作を行います。

- a. [詳細] で、コラボレーションの [名前] と [説明] を入力します。

この情報は、コラボレーションに参加するよう招待されたコラボレーションメンバーに表示されます。メンバーは、[名前] と [説明] でコラボレーションが何を指しているのかを把握できます。

- b. [メンバー] で以下の操作を行います。

- i. [メンバー 1: 自分] の [メンバー表示名] に、コラボレーションで表示する自身のメンバー表示名を入力します。

Note

ID はメンバー AWS アカウント ID に自動的に含まれます。 AWS アカウント

- ii. メンバー 2 には、コラボレーションに招待するメンバーの表示名とメンバー AWS アカウント ID を入力します。

メンバー表示名とメンバー AWS アカウント ID は、コラボレーションに招待されたすべてのユーザーに表示されます。これらのフィールドに入力して保存した値を、後から編集することはできません。

Note

コラボレーションメンバーのメンバー AWS アカウント ID とメンバー表示名は、コラボレーションに参加している招待されたアクティブなコラボレーター全員に表示されるため、この点について本人に知らせておく必要があります。

- iii. 別のメンバーを追加する場合は、[別のメンバーを追加] を選択します。次に、コラボレーションに招待するデータを提供できる各メンバーのメンバー表示名とメンバー AWS アカウント ID を入力します。
- c. 分析ログ記録を有効にする場合は、分析ログ記録を有効にするチェックボックスをオンにします。
 - サポートされているログタイプのクエリからのログチェックボックスをオンにします。

Amazon CloudWatch Logs アカウントの SQL クエリから生成されたログを受け取ります。

- d. 詳細モニタリングを有効にする場合は、詳細モニタリングを有効にするチェックボックスをオンにします。

分析ランナーと設定済み支払者は、メンバーシップの作成時に詳細なメトリクスを有効にすることを選択できます。有効にすると、クエリのパフォーマンスやリソース使用率など、コラボレーションの運用モニタリングのために詳細なモニタリングメトリクスが CloudWatch に発行されます。これらのメトリクスは、分析ランナーと設定済みの支払者がそれぞれの AWS アカウントで使用できます。

CloudWatch の料金の詳細については、[CloudWatch の料金](#)をご覧ください。

- e. 「許可されたクエリ結果リージョン」で、クエリ結果を送信する AWS リージョン 1 つ以上の を選択します。

デフォルトでは、現在のリージョン (バージニア北部 us-east-1 など) のみが選択されます。

 Important

クロスリージョンクエリ結果の配信を有効にすると、結果が処理され、ソースリージョンの外部に保存される場合があります。

リージョンの詳細については、の「[リージョンとエンドポイント](#)」を参照してくださいAWS 全般のリファレンス。

- f. (オプション) 変更リクエストの手動承認なしで自動的に変更できる設定を設定することで、自動変更リクエスト承認でデータへのアクセスを管理します。デフォルトでは、一部の設定は変更リクエストを送信することによってのみ変更できます。変更リクエストを有効にするには、すべてのメンバーが承認する必要があります。
- メンバーの能力を付与する – 手動の承認なしでコラボレーションメンバーに付与できる能力を選択します。メンバーはいつでもデータを提供できます。
 - 機能を選択します。
 - 寄与データ (常に有効)
 - 結果を受け取る
 - これらの機能を持つ新しいメンバーの自動承認 – 許可されている場合、上記で選択した機能で追加されたメンバーはすぐにコラボレーションに参加します。他の機能で追加されたメンバーは、参加するために手動で承認する必要があります。
 - 自動的に取り消すことができる機能 - 手動承認なしで取り消すことができる機能を選択します。メンバーはいつでもデータを提供できます。
 - 機能を選択します。
 - 寄与データ (常に有効)
 - 結果を受け取る

このオプションを選択すると、コラボレーションの詳細ページの 詳細 タブにある変更リクエスト履歴を通じて、すべてのコラボレーション設定の変更を追跡できます。

- g. (オプション) 暗号化コンピューティング機能を有効にする場合は、暗号化コンピューティングを有効にするチェックボックスをオンにします。

- i. 次の暗号化カバレッジパラメータを選択します。

- plaintext列を許可

完全に暗号化されたテーブルが必要な場合は、いいえ を選択します。

暗号化されたテーブルでcleartext列を許可する場合は、[はい] を選択します。

特定の列で SUM または AVG を実行するには、その列がcleartextである必要があります。

- NULL 値を保存

NULL 値を保持しない場合は [いいえ] を選択します。NULL 値は暗号化されたテーブルで NULL として表示されません。

NULL 値を保持する場合は [はい] を選択します。NULL 値は暗号化されたテーブルで NULL として表示されます。

- ii. 次のフィンガープリントパラメータを選択します。

- 重複を許可

1 つのfingerprint列で重複するエントリを許可しない場合は、[いいえ] を選択します。

1 つのfingerprint列で重複するエントリを許可する場合は、[はい] を選択します。

- 名前の異なる列の JOIN を許可

名前の異なるfingerprint列を結合しない場合は、[いいえ] を選択します。

名前の異なるfingerprint列を結合する場合は、[はい] を選択します。

暗号コンピューティングパラメータの詳細については、「[暗号コンピューティングパラメータ](#)」を参照してください。

で使用するデータを暗号化する方法の詳細については AWS Clean Rooms、「」を参照してください [Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)。

 Note

次のステップを完了する前に、これらの設定を注意深く確認してください。
コラボレーションを作成したら、コラボレーション名、説明、ログが Amazon CloudWatch Logs に保存されているかどうかのみを編集できます。

- h. コラボレーションリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
 - i. [次へ] を選択します。
5. ステップ 2: クエリとジョブを使用した分析のメンバー機能を指定するには、サポートされている分析タイプでクエリチェックボックスをオンのままにして、目標に基づいて推奨アクションを実行します。

目標	推奨されるアクション
コラボレーション内のデータにクエリを実行し、結果を受け取る	- クエリを実行するメンバーとして自分自身を選択します。 - ドロップダウンリストから分析から結果を受け取ることができるメンバーとして自分自身を選択します。
自分がコラボレーション内のデータにクエリを実行し、結果の受け取りは別のメンバーに割り当てる	- クエリを実行するメンバーとして自分自身を選択します。 - ドロップダウンリストから分析から結果を受信できるメンバーを選択します。
自分がコラボレーションでのクエリ結果を受け取り、データに対するクエリの実行は別のメンバーに割り当てる	- ドロップダウンリストから、クエリを実行するメンバーを選択します。 - ドロップダウンリストから分析から結果を受け取ることができるメンバーとして自分自身を選択します。

目標	推奨されるアクション
自分がコラボレーションを作成して管理し、データに対するクエリの実行と結果の受け取りはそれぞれ別のメンバーに割り当てる	1. ドロップダウンリストから、クエリを実行するメンバーを選択します。 2. ドロップダウンリストから分析から結果を受信できるメンバーを選択します。

- a. Clean Rooms ML を使用している場合、専用ワークフローを使用した ML モデリングには、
 - i. (オプション) ドロップダウンリストから、トレーニング済みモデルから出力を受け取ることができるメンバーを選択します。
 - ii. (オプション) ドロップダウンリストからモデル推論から出力を受信できるメンバーを選択します。
 - b. を使用して ID 解決 AWS Entity Resolution でメンバー機能を表示します。
 - c. [次へ] を選択します。
6. ステップ 3: 支払いを設定する場合、クエリを使用した分析では、目標に基づいて次のいずれかのアクションを実行します。

目標	推奨されるアクション
クエリを実行するメンバーを、クエリの計算コストを負担するメンバーとして割り当てる	1. クエリを使用した分析では、クエリに対して支払うメンバーを、クエリを実行できるメンバーと同じメンバーとして選択します。 2. [次へ] を選択します。
クエリの計算コストの負担を別のメンバーを割り当てる	1. クエリを使用した分析では、クエリの支払いを行うメンバーとして自分自身を選択します。 2. [次へ] を選択します。

専用ワークフローを使用した ML モデリングの場合、設定された類似モデルの作成者は類似モデリングに対して支払いを行うメンバーです。

を使用した ID 解決 AWS Entity Resolution の場合、ID マッピングテーブルの作成者は、ID マッピングテーブルの支払いを行うメンバーです。

7. ステップ 4: メンバーシップを設定するには、次のいずれかのオプションを選択します。

Yes, join by creating membership now

1. 結果設定のデフォルトの場合、クエリ結果設定の場合、結果を受信できるメンバーである場合、
 - a. Amazon S3 の結果の送信先については、Amazon S3 の送信先を入力するか、S3 の参照を選択して S3 バケットを選択します。
 - b. [結果フォーマット] で、[CSV] または [Parquet] を選択します。
 - c. (Spark のみ) 結果ファイルで、複数または単一を選択します。
 - d. (オプション) サービスアクセスで、S3 送信先に最大 24 時間かかるクエリを配信する場合は、サービスロールの追加を選択して、チェックボックスをオンにするまでに最大 24 時間かかるクエリをサポートします。

完了までに最大 24 時間かかる大規模なクエリは、S3 送信先に配信されます。

チェックボックスをオンにしない場合、12 時間以内に完了したクエリのみが S3 の場所に配信されます。

- e. [新しいサービスロールを作成して使用] または [既存のサービスロールを使用] を選択して、[サービスアクセス] 許可を指定します。

... を選択した場合	結果
新しいサービスロールを作成して使用	• AWS Clean Rooms は、このテーブルに必要なポリシーを持つサービスロールを作成します。 • デフォルトの [サービスロール名] は <code>cleanrooms-result-receiver-<timestamp></code> です。 • ロールを作成してポリシーをアタッチするアクセス許可が必要です。
既存のサービスロールを使用	i. ドロップダウンリストから [既存のサービスロール名] を選択します。

... を選択した場合	結果
	ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。 ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。 ii. [IAM で表示] 外部リンクを選択してサービスロールを表示します。 既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。 デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

 Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、「」を参照してください[AWS の 管理ポリシー AWS Clean Rooms](#)。
- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。
- ロールポリシーを変更できない場合は、 がサービスロールのポリシーを見つけ AWS Clean Rooms られなかったことを示すエラーメッセージが表示されます。

2. ログ設定で、Amazon CloudWatch Logs のログストレージに次のいずれかのオプションを選択します。

 Note

クエリログ記録を有効にすることを選択した場合は、ログ設定セクションが表示されます。

- a. オンにすると、関連するクエリログが Amazon CloudWatch Logs アカウントに保存されます。

各メンバーは、自分が開始したクエリ、または自分のデータを含むクエリのログのみを受け取ることができます。

また、結果を受け取ることができるメンバーは、自分のデータがクエリで利用されていない場合でも、コラボレーションで実行されたすべてのクエリのログを受け取ります。

サポートされているログタイプでは、クエリログのチェックボックスがデフォルトでオンになっています。

 Note

[クエリログ記録] を有効にした後、ログストレージが設定されて Amazon CloudWatch Logs でログの受信が開始されるまでに数分かかることがあります。この短い間、クエリを行えるメンバーがクエリを実行しても、実際にはログが送信されない場合があります。

- b. 無効にすると、関連するクエリログは Amazon CloudWatch Logs アカウントに保存されません。
3. メンバーシップリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
4. クエリコンピューティングに料金を支払うメンバーの場合は、このコラボレーションチェックボックスでコンピューティングコストの支払いに同意する を選択して、承諾を示します。

 Note

続行するには、このチェックボックスをオンにする必要があります。
料金の計算方法の詳細については、「[の料金 AWS Clean Rooms](#)」を参照してください。

[クエリコンピューティングコストを負担するメンバー](#)で、[クエリを行えるメンバー](#)でない場合は、AWS Budgets を使用して の予算を設定し、最大予算に達したら通知 AWS Clean Rooms を受け取ることをお勧めします。詳細については、「AWS Cost Management ユーザーガイド」の「[AWS Budgetsを用いてコストを管理する](#)」を参照してください。通知の設定の詳細については、「AWS Cost Management ユーザーガイド」の「[予算の通知に関する Amazon SNS トピックを作成する](#)」を参照してください。予算の上限に達した場合は、クエリを行えるメンバーに連絡するか、[コラボレーションから退出](#)できます。コラボレーションから退出すると、それ以上クエリの実行は許可されなくなるため、クエリの計算コストも請求されなくなります。

5. [次へ] を選択します。

コラボレーションとメンバーシップの両方が作成されます。

コラボレーションのステータスはアクティブになります。

No, I will create a membership later

1. [次へ] を選択します。

コラボレーションのみが作成されます。

コラボレーションのステータスは非アクティブになります。

8. ステップ 5: 確認して作成するには、以下を実行します。

- a. 前のステップで行った選択内容を確認し、必要に応じて編集します。
- b. いずれかのオプションを選択します。

前のステップでの選択内容	選択内容
コラボレーションと共にメンバーシップを作成 ([はい、今すぐメンバーシップを作成してご入会ください])	コラボレーションとメンバーシップを作成
コラボレーションは作成するが、現時点ではメンバーシップを作成しない ([いいえ、後でメンバーシップを作成します])	コラボレーションを作成

コラボレーションが正常に作成されると、[コラボレーション] の下にコラボレーションの詳細ページが表示されます。

これで次の作業に進むことができます。

- [で分析するデータテーブルを準備します AWS Clean Rooms](#)。(独自のイベントデータを分析する場合、または ID データをクエリする場合はオプション)。
- [設定済みテーブルをコラボレーションに関連付ける](#) (独自のイベントデータを分析する場合はオプション)。
- [設定済みテーブルの分析ルールを追加します](#)。(独自のイベントデータを分析する場合はオプション)。
- [メンバーシップを作成してコラボレーションに参加する](#) (メンバーシップを既に作成している場合は省略可能)
- [メンバーをコラボレーションに招待します](#)。

クエリとジョブのコラボレーションの作成

この手順では、[コラボレーション作成者](#)として次のタスクを実行します。

- [コラボレーションを作成します](#)。
- 1 人以上の[メンバー](#)を[コラボレーション](#)に招待します。
- [クエリやジョブを実行できる](#)メンバーや、[結果を受け取ることができるメンバー](#)など、メンバーに権限を割り当てます。

コラボレーション作成者が結果を受け取ることができるメンバーでもある場合は、結果の送信先と形式を指定します。また、結果の送信先に結果を書き込むためのサービスロール Amazon リソースネーム (ARN) も提供します。

- [コラボレーションでクエリとジョブのコンピューティングコストを負担するメンバー](#)を設定します。

開始する前に、以下の前提条件を満たしていることを確認してください。

- コラボレーションに招待する各メンバーの名前と AWS アカウント ID があります。
- 各メンバーの名前と AWS アカウント ID をコラボレーションのすべてのメンバーと共有するためのアクセス許可があります。

 Note

コラボレーションの作成後にメンバーを追加することはできません。

AWS SDKs を使用してコラボレーションを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

クエリとジョブのコラボレーションを作成するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエーターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 右上隅にある [コラボレーションを作成] を選択します。
4. [ステップ 1: コラボレーションの定義] で以下の操作を行います。

- a. [詳細] で、コラボレーションの [名前] と [説明] を入力します。

この情報は、コラボレーションに参加するよう招待されたコラボレーションメンバーに表示されます。メンバーは、[名前] と [説明] でコラボレーションが何を指しているのかを把握できます。

- b. [メンバー] で以下の操作を行います。

- i. [メンバー 1: 自分] の [メンバー表示名] に、コラボレーションで表示する自身のメンバー表示名を入力します。

Note

ID はメンバー AWS アカウント ID に自動的に含まれます。 AWS アカウント

- ii. メンバー 2 には、コラボレーションに招待するメンバーの表示名とメンバー AWS アカウント ID を入力します。

メンバー表示名とメンバー AWS アカウント ID は、コラボレーションに招待されたすべてのユーザーに表示されます。これらのフィールドに入力して保存した値を、後から編集することはできません。

Note

コラボレーションメンバーのメンバー AWS アカウント ID とメンバー表示名は、コラボレーションに参加している招待されたアクティブなコラボレーター全員に表示されるため、この点について本人に知らせておく必要があります。

- iii. 別のメンバーを追加する場合は、[別のメンバーを追加] を選択します。次に、コラボレーションに招待するデータを提供できる各メンバーのメンバー表示名とメンバー AWS アカウント ID を入力します。
- c. 分析ログ記録を有効にする場合は、分析ログ記録を有効にするチェックボックスを選択し、サポートされているログタイプを選択します。
 - SQL クエリから生成されたログを受信する場合は、クエリからのログチェックボックスをオンにします。
 - PySpark を使用してジョブから生成されたログを受信する場合は、ジョブからのログチェックボックスをオンにします。
- d. 詳細モニタリングを有効にする場合は、詳細モニタリングを有効にするチェックボックスをオンにします。

分析ランナーと設定済み支払者は、メンバーシップの作成時に詳細なメトリクスを有効にすることを選択できます。有効にすると、クエリのパフォーマンスやリソース使用率など、コラボレーションの運用モニタリングのために詳細なモニタリングメトリクスが CloudWatch に発行されます。これらのメトリクスは、分析ランナーと設定済みの支払者がそれぞれの AWS アカウントで使用できます。

CloudWatch の料金の詳細については、[CloudWatch の料金](#)をご覧ください。

- e. 「許可されたクエリ結果リージョン」で、クエリ結果を送信する AWS リージョン 1 つ以上の を選択します。

デフォルトでは、現在のリージョン (バージニア北部 us-east-1 など) のみが選択されます。

 Important

クロスリージョンクエリ結果の配信を有効にすると、結果が処理され、ソースリージョンの外部に保存される場合があります。

リージョンの詳細については、の「[リージョンとエンドポイント](#)」を参照してくださいAWS 全般のリファレンス。

- f. (オプション) 変更リクエストの手動承認なしで自動的に変更できる設定を設定することで、自動変更リクエスト承認でデータへのアクセスを管理します。デフォルトでは、一部の設定は変更リクエストを送信することによってのみ変更できます。変更リクエストを有効にするには、すべてのメンバーが承認する必要があります。
- メンバーの権限を付与する – 手動承認なしでコラボレーションメンバーに付与できる機能を選択します。メンバーはいつでもデータを提供できます。
 - 機能を選択します。
 - 寄与データ (常に有効)
 - 結果を受け取る
 - これらの機能を持つ新しいメンバーを自動承認する – 許可されている場合、上記で選択した機能で追加されたメンバーはすぐにコラボレーションに参加します。他の機能で追加されたメンバーは、参加を手動で承認する必要があります。
 - 自動的に取り消すことができる機能 - 手動承認なしで取り消すことができる機能を選択します。メンバーはいつでもデータを提供できます。
 - 機能を選択します。
 - 寄与データ (常に有効)
 - 結果を受け取る

このオプションを選択すると、コラボレーションの詳細ページの 詳細 タブにある変更リクエスト履歴を通じて、すべてのコラボレーション設定の変更を追跡できます。

- g. (オプション) 暗号化コンピューティング機能を有効にする場合は、暗号化コンピューティングを有効にするチェックボックスをオンにします。

- i. 次の暗号化カバレッジパラメータを選択します。

- plaintext列を許可

完全に暗号化されたテーブルが必要な場合は、いいえ を選択します。

暗号化されたテーブルでcleartext列を許可する場合は、[はい] を選択します。

特定の列で SUM または AVG を実行するには、その列がcleartextである必要があります。

- NULL 値を保存

NULL 値を保持しない場合は [いいえ] を選択します。NULL 値は暗号化されたテーブルで NULL として表示されません。

NULL 値を保持する場合は [はい] を選択します。NULL 値は暗号化されたテーブルで NULL として表示されます。

- ii. 次のフィンガープリントパラメータを選択します。

- 重複を許可

1 つのfingerprint列で重複するエントリを許可しない場合は、[いいえ] を選択します。

1 つのfingerprint列で重複するエントリを許可する場合は、[はい] を選択します。

- 名前の異なる列の JOIN を許可

名前の異なるfingerprint列を結合しない場合は、[いいえ] を選択します。

名前の異なるfingerprint列を結合する場合は、[はい] を選択します。

暗号コンピューティングパラメータの詳細については、「[暗号コンピューティングパラメータ](#)」を参照してください。

で使用するデータを暗号化する方法の詳細については AWS Clean Rooms、「」を参照してください [Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)。

 Note

次のステップを完了する前に、これらの設定を注意深く確認してください。コラボレーションを作成したら、コラボレーション名、説明、およびログが Amazon CloudWatch Logs に保存されているかどうかのみを編集できます。

- h. コラボレーションリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
 - i. [次へ] を選択します。
5. ステップ 2: メンバー機能を指定するには、以下を実行します。
- a. クエリとジョブを使用した分析では、サポートされている分析タイプで、ジョブチェックボックスをオンにします。

クエリチェックボックスはデフォルトで選択されています。

- i. ドロップダウンリストからクエリとジョブを実行できるメンバーを選択します。
- ii. ドロップダウンリストから分析から結果を受け取ることができるメンバーを選択します。

 Note

PySpark 分析テンプレートを作成するメンバーも、結果を受け取るメンバーである必要があります。

- b. Clean Rooms ML を使用している場合、専用ワークフローを使用した ML モデリングには、
 - i. (オプション) ドロップダウンリストから、トレーニング済みモデルからの出力を受信できるメンバーを選択します。
 - ii. (オプション) ドロップダウンリストからモデル推論から出力を受信できるメンバーを選択します。
 - c. を使用して ID 解決 AWS Entity Resolution でメンバー機能を表示します。
 - d. [次へ] を選択します。
6. ステップ 3: 支払いを設定する

- a. クエリとジョブを使用した分析では、クエリとジョブの支払いを行うメンバーを選択します。

クエリとジョブを実行できるメンバーを、クエリとジョブの計算コストを支払うメンバーとして割り当てることができます。

クエリとジョブのコンピューティングコストを支払うために、別のメンバーを割り当てることができます。

- b. 専用ワークフローを使用した ML モデリングの場合、設定された類似モデルの作成者は類似モデリングに対して支払いを行うメンバーです。
 - c. を使用した ID 解決 AWS Entity Resolution の場合、ID マッピングテーブルの作成者は、ID マッピングテーブルの支払いを行うメンバーです。
 - d. [次へ] を選択します。
7. ステップ 4: メンバーシップを設定するには、次のいずれかのオプションを選択します。

Yes, join by creating membership now

1. 結果設定のデフォルトの場合、クエリ結果設定の場合、結果を受信できるメンバーである場合、
 - a. クエリのデフォルト設定を設定するチェックボックスをオンにします。Amazon S3 の結果の送信先については、Amazon S3 の送信先を入力するか、S3 の参照を選択して S3 バケットを選択します。
 - b. [結果フォーマット] で、[CSV] または [Parquet] を選択します。
 - c. (Spark のみ) 結果ファイルで、複数または単一を選択します。
 - d. (オプション) サービスアクセスで、S3 送信先に最大 24 時間かかるクエリを配信する場合は、サービスロールを追加して、最大 24 時間かかるクエリをサポートするチェックボックスをオンにします。

完了までに最大 24 時間かかる大規模なクエリは、S3 送信先に配信されます。

チェックボックスをオンにしない場合、12 時間以内に完了したクエリのみが S3 の場所に配信されます。

- e. [新しいサービスロールを作成して使用] または [既存のサービスロールを使用] を選択して、[サービスアクセス] 許可を指定します。

... を選択した場合	結果
新しいサービスロールを作成して使用	• AWS Clean Rooms は、このテーブルに必要なポリシーを持つサービスロールを作成します。 • デフォルトの [サービスロール名] は <code>cleanrooms-result-receiver-<timestamp></code> です。 • ロールを作成してポリシーをアタッチするアクセス許可が必要です。
既存のサービスロールを使用	i. ドロップダウンリストから [既存のサービスロール名] を選択します。 ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。 ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。 ii. [IAM で表示] 外部リンクを選択してサービスロールを表示します。 既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。 デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

 Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、「」を参照してください [AWS の 管理ポリシー AWS Clean Rooms](#)。
- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。
- ロールポリシーを変更できない場合は、 がサービスロールのポリシーを見つけ AWS Clean Rooms られなかったことを示すエラーメッセージが表示されます。

2. ジョブの結果については、

Example

例: **s3://bucket/prefix**

- a. ジョブのデフォルト設定を設定するチェックボックスを選択し、SAmazon S3S3 の結果の送信先を指定するか、S3 を参照を選択して使用可能な S3S3 バケットのリストから選択します。
 - b. ドロップダウンリストから既存のサービスロール名を選択して、サービスアクセス許可を指定します。
3. ログ設定で、Amazon CloudWatch Logs のログストレージに次のいずれかのオプションを選択します。

 Note

クエリログ記録を有効にすることを選択した場合は、ログ設定セクションが表示されます。

- a. オンにすると、関連するクエリログが Amazon CloudWatch Logs アカウントに保存されます。

各メンバーは、自分が開始したクエリ、または自分のデータを含むクエリのログのみを受け取ることができます。

また、結果を受け取ることができるメンバーは、自分のデータがクエリで利用されていない場合でも、コラボレーションで実行されたすべてのクエリのログを受け取ります。

サポートされているログタイプで、コラボレーション作成者がサポート対象として選択したログタイプから選択します。

サポートされているログタイプでは、クエリログとジョブログのチェックボックスがデフォルトでオンになっています。

 Note

分析ログ記録を有効にすると、ログストレージがセットアップされ、Amazon CloudWatch Logs でログの受信が開始されるまでに数分かかることがあります。この短い間、クエリを行えるメンバーがクエリを実行しても、実際にはログが送信されない場合があります。

- b. 無効にすると、関連するクエリログは Amazon CloudWatch Logs アカウントに保存されません。
4. メンバーシップリソースのメンバーシップタグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
5. クエリコンピューティング、ジョブコンピューティング、またはその両方に料金を支払っているメンバーの場合は、このコラボレーションチェックボックスでコンピューティングコストの支払いに同意するを選択します。

 Note

続行するには、このチェックボックスをオンにする必要があります。
料金の計算方法の詳細については、「[の料金 AWS Clean Rooms](#)」を参照してください。

クエリコンピューティングコストを支払っているが、クエリを行えるメンバーでない場合は、 を使用して予算 AWS Budgets を設定し、最大予算に達したら通知 AWS Clean Rooms を受け取ることをお勧めします。詳細については、「AWS Cost Management

ユーザーガイド」の「[AWS Budgetsを用いてコストを管理する](#)」を参照してください。通知の設定の詳細については、「AWS Cost Management ユーザーガイド」の「[予算の通知に関する Amazon SNS トピックを作成する](#)」を参照してください。予算の上限に達した場合は、クエリを行えるメンバーに連絡するか、[コラボレーションから退出](#)できます。コラボレーションから退出すると、それ以上クエリの実行は許可されなくなるため、クエリの計算コストも請求されなくなります。

6. [次へ] を選択します。

コラボレーションとメンバーシップの両方が作成されます。

コラボレーションのステータスはアクティブになります。

No, I will create a membership later

1. [次へ] を選択します。

コラボレーションのみが作成されます。

コラボレーションのステータスは非アクティブになります。

8. ステップ 5: 確認して作成するには、次の手順を実行します。

- a. 前のステップで行った選択内容を確認し、必要に応じて編集します。
- b. いずれかのオプションを選択します。

前のステップでの選択内容	選択内容
コラボレーションと共にメンバーシップを作成 ([はい、今すぐメンバーシップを作成してご入会ください])	コラボレーションとメンバーシップを作成
コラボレーションは作成するが、現時点ではメンバーシップを作成しない ([いいえ、後でメンバーシップを作成します])	コラボレーションを作成

コラボレーションが正常に作成されると、[コラボレーション] の下にコラボレーションの詳細ページが表示されます。

これで次の作業に進むことができます。

- [で分析するデータテーブルを準備します AWS Clean Rooms](#)。(独自のイベントデータを分析する場合、または ID データをクエリする場合はオプション)。
- [設定済みテーブルをコラボレーションに関連付ける](#) (独自のイベントデータを分析する場合はオプション)。
- [設定済みテーブルの分析ルールを追加します](#)。(独自のイベントデータを分析する場合はオプション)。
- [メンバーシップを作成してコラボレーションに参加する](#) (メンバーシップを既に作成している場合は省略可能)
- [メンバーをコラボレーションに招待します](#)。

ML モデリング用のコラボレーションの作成

この手順では、[コラボレーション作成者](#)として次のタスクを実行します。

- [コラボレーションを作成します](#)。
- 1 人以上の [メンバー](#) を [コラボレーション](#) に招待します。
- などのメンバーに機能を割り当てる
 - [クエリを実行できるメンバー](#)
 - [結果を受け取ることができるメンバー](#)
 - トレーニング済みモデルから出力を受け取ることができるメンバー
 - モデル推論から出力できるメンバー

コラボレーション作成者が結果を受け取ることができるメンバーでもある場合は、結果の送信先と形式を指定します。また、結果の送信先に結果を書き込むためのサービスロール Amazon リソースネーム (ARN) も提供します。

- [コラボレーションにおけるコンピューティングコスト、モデルトレーニング、モデル推論コストの支払いを担当するメンバー](#)を設定します。

開始する前に、以下の前提条件を満たしていることを確認してください。

- コラボレーションに招待する各メンバーの名前と AWS アカウント ID があります。
- 各メンバーの名前と AWS アカウント ID をコラボレーションのすべてのメンバーと共有するためのアクセス許可があります。

Note

コラボレーションの作成後にメンバーを追加することはできません。

AWS SDKs を使用してコラボレーションを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

ML モデリングのコラボレーションを作成するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエーターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 右上隅にある [コラボレーションを作成] を選択します。
4. [ステップ 1: コラボレーションの定義] で以下の操作を行います。

- a. [詳細] で、コラボレーションの [名前] と [説明] を入力します。

この情報は、コラボレーションに参加するよう招待されたコラボレーションメンバーに表示されます。メンバーは、[名前] と [説明] でコラボレーションが何を指しているのかを把握できます。

- b. [メンバー] で以下の操作を行います。

- i. [メンバー 1: 自分] の [メンバー表示名] に、コラボレーションで表示する自身のメンバー表示名を入力します。

Note

ID はメンバー AWS アカウント ID に自動的に含まれます。 AWS アカウント

- ii. メンバー 2 には、コラボレーションに招待するメンバーの表示名とメンバー AWS アカウント ID を入力します。

メンバー表示名とメンバー AWS アカウント ID は、コラボレーションに招待されたすべてのユーザーに表示されます。これらのフィールドに入力して保存した値を、後から編集することはできません。

Note

コラボレーションメンバーのメンバー AWS アカウント ID とメンバー表示名は、コラボレーションに参加している招待されたアクティブなコラボレーター全員に表示されるため、この点について本人に知らせておく必要があります。

iii. 別のメンバーを追加する場合は、[別のメンバーを追加] を選択します。次に、コラボレーションに招待するデータを提供できる各メンバーのメンバー表示名とメンバー AWS アカウント ID を入力します。

- c. 分析ログ記録を有効にする場合は、分析ログ記録を有効にするチェックボックスを選択し、サポートされているログタイプでクエリからログを選択します。
- d. 詳細モニタリングを有効にする場合は、詳細モニタリングを有効にするチェックボックスをオンにします。

分析ランナーと設定済み支払者は、メンバーシップの作成時に詳細なメトリクスを有効にすることを選択できます。有効にすると、クエリのパフォーマンスやリソース使用率など、コラボレーションの運用モニタリングのために詳細なモニタリングメトリクスが CloudWatch に発行されます。これらのメトリクスは、分析ランナーと設定済みの支払者がそれぞれの AWS アカウントで使用できます。

CloudWatch の料金の詳細については、[CloudWatch の料金](#)をご覧ください。

- e. 「許可されたクエリ結果リージョン」で、クエリ結果を送信する AWS リージョン 1 つ以上の を選択します。

デフォルトでは、現在のリージョン (バージニア北部 us-east-1 など) のみが選択されます。

Important

クロスリージョンクエリ結果の配信を有効にすると、結果が処理され、ソースリージョンの外部に保存される場合があります。

リージョンの詳細については、の「[リージョンとエンドポイント](#)」を参照してくださいAWS 全般のリファレンス。

- f. (オプション) 変更リクエストの手動承認なしで自動的に変更できる設定を設定することで、自動変更リクエスト承認でデータへのアクセスを管理します。デフォルトでは、一部の設定は変更リクエストを送信することによってのみ変更できます。変更リクエストを有効にするには、すべてのメンバーが承認する必要があります。
- メンバーの権限を付与する – 手動承認なしでコラボレーションメンバーに付与できる機能を選択します。メンバーはいつでもデータを提供できます。
 - 機能を選択します。
 - 寄与データ (常に有効)
 - 結果を受け取る
 - これらの機能を持つ新しいメンバーの自動承認 – 許可されている場合、上記で選択した機能で追加されたメンバーはすぐにコラボレーションに参加します。他の機能で追加されたメンバーは、参加を手動で承認する必要があります。
 - 自動的に取り消すことができる機能 - 手動承認なしで取り消すことができる機能を選択します。メンバーはいつでもデータを提供できます。
 - 機能を選択します。
 - 寄与データ (常に有効)
 - 結果を受け取る

このオプションを選択すると、コラボレーションの詳細ページの 詳細 タブにある変更リクエスト履歴を通じて、すべてのコラボレーション設定の変更を追跡できます。

- g. (オプション) 暗号化コンピューティング機能を有効にする場合は、暗号化コンピューティングを有効にするチェックボックスをオンにします。

- i. 次の暗号化カバレッジパラメータを選択します。

- plaintext列を許可

完全に暗号化されたテーブルが必要な場合は、いいえ を選択します。

暗号化されたテーブルでcleartext列を許可する場合は、[はい] を選択します。

特定の列で SUM または AVG を実行するには、その列がcleartextである必要があります。

- NULL 値を保存

NULL 値を保持しない場合は [いいえ] を選択します。NULL 値は暗号化されたテーブルで NULL として表示されません。

NULL 値を保持する場合は [はい] を選択します。NULL 値は暗号化されたテーブルで NULL として表示されます。

ii. 次のフィンガープリントパラメータを選択します。

- 重複を許可

1 つの fingerprint 列で重複するエントリを許可しない場合は、[いいえ] を選択します。

1 つの fingerprint 列で重複するエントリを許可する場合は、[はい] を選択します。

- 名前の異なる列の JOIN を許可

名前の異なる fingerprint 列を結合しない場合は、[いいえ] を選択します。

名前の異なる fingerprint 列を結合する場合は、[はい] を選択します。

暗号コンピューティングパラメータの詳細については、「[暗号コンピューティングパラメータ](#)」を参照してください。

で使用するデータを暗号化する方法の詳細については AWS Clean Rooms、「」を参照してください [Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)。

 Note

次のステップを完了する前に、これらの設定を注意深く確認してください。コラボレーションを作成したら、コラボレーション名、説明、ログが Amazon CloudWatch Logs に保存されているかどうかのみを編集できます。

h. コラボレーションリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。

i. [次へ] を選択します。

5. ステップ 2: メンバーの能力を指定するには、

a. クエリとジョブを使用した分析では、サポートされている分析タイプで、クエリチェックボックスをオンにしたままにします。

- b. クエリを実行するには、モデルトレーニングを開始するメンバーを選択します。
 - c. 分析から結果を受け取る で、クエリ結果を受け取るメンバーを 1 人以上選択します。
 - d. 専用ワークフローを使用した ML モデリングの場合、
 - i. トレーニング済みモデルからの出力を受信する では、モデルアーティファクトやメトリクスなど、トレーニング済みモデル結果を受信するメンバーを選択します。
 - ii. モデル推論から出力を受信する で、モデル推論結果を受け取るメンバーを選択します。
 - e. を使用して ID 解決 AWS Entity Resolutionでメンバー機能を表示します。
6. ステップ 3: 支払いを設定する
- a. クエリを使用した分析で、クエリの支払いについて、次のいずれかのアクションを実行します。
 - 同じメンバーに対して支払いを行い、クエリを実行するには、クエリの実行に選択したのと同じメンバーを選択します。
 - 別のメンバーにクエリコストの支払いを依頼するには、メンバーアカウントを選択します。
 - b. 専用ワークフローを使用した ML モデリングの場合、
 - モデルトレーニングに支払うメンバーを選択します。
 - c. 推論ジョブの支払いを行うメンバーを選択します。
 - d. Pay for Lookalike モデリングの場合、アクションは必要ありません。設定された類似モデルの作成者は、類似モデリングに対して料金を支払うメンバーです。
 - e. (オプション) Synthetic データ生成に支払うメンバーを選択します。
 - f. を使用した ID 解決 AWS Entity Resolutionの場合、アクションは必要ありません。ID マッピングテーブルの作成者は、ID マッピングテーブルの支払いを行うメンバーです。
7. [次へ] を選択します。
8. ステップ 4: メンバーシップを設定するには、コラボレーションメンバーシップで、次のいずれかのオプションを選択します。

Yes, join by creating membership now

1. 結果設定のデフォルトの場合、クエリ結果設定の場合、結果を受信できるメンバーである場合、

- a. クエリのデフォルト設定を設定するチェックボックスをオンにします。
- b. Amazon S3 の結果の送信先については、Amazon S3 の送信先を入力するか、S3 の参照を選択して S3 バケットを選択します。
- c. [結果フォーマット] で、[CSV] または [Parquet] を選択します。
- d. (Spark のみ) 結果ファイルの場合は、複数または単一を選択します。
- e. (オプション) S3 送信先に最大 24 時間かかるクエリを配信する場合は、サービスロールを追加を選択して、チェックボックスをオンにするまでに最大 24 時間かかるクエリをサポートします。

完了までに最大 24 時間かかる大規模なクエリは、S3 送信先に配信されます。

チェックボックスをオンにしない場合、12 時間以内に完了したクエリのみが S3 の場所に配信されます。

- f. [新しいサービスロールを作成して使用] または [既存のサービスロールを使用] を選択して、[サービスアクセス] 許可を指定します。

... を選択した場合	結果
新しいサービスロールを作成して使用	• AWS Clean Rooms は、このテーブルに必要なポリシーを持つサービスロールを作成します。 • デフォルトの [サービスロール名] は <code>cleanrooms-result-receiver-<timestamp></code> です。 • ロールを作成してポリシーをアタッチするアクセス許可が必要です。

... を選択した場合	結果
既存のサービスロールを使用	i. ドロップダウンリストから [既存のサービスロール名] を選択します。 ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。 ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。 ii. [IAM で表示] 外部リンクを選択してサービスロールを表示します。 既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。 デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

 Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、「」を参照してください[AWS の 管理ポリシー AWS Clean Rooms](#)。
- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。

- ロールポリシーを変更できない場合は、がサービスロールのポリシーを見つけ AWS Clean Rooms られなかったことを示すエラーメッセージが表示されます。

2. ML 設定の場合、

- a. ML 設定の作成チェックボックスを選択し、S3 送信先を入力して Amazon S3 のモデル出力送信先を指定するか、SS3を選択して使用可能な SS3 バケットのリストから選択します。S3
 - b. 新しいサービスロールを作成して使用するか、既存のサービスロールを使用するかを選択して、サービスアクセス許可を指定します。
 - c. S3 バケットが暗号化されている場合は、送信先バケットが KMS キーで暗号化されているチェックボックスを選択し、を入力するAWS KMS keyか、の作成 AWS KMS keyを選択して新しい KMS キーを作成します。
3. メンバーシップリソースのメンバーシップタグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
 4. クエリコンピューティングに料金を支払っているメンバーの場合は、このコラボレーションチェックボックスでコンピューティングコストの支払いに同意する を選択して、承諾を示します。

Note

続行するには、このチェックボックスをオンにする必要があります。
料金の計算方法の詳細については、「[の料金 AWS Clean Rooms](#)」を参照してください。

[クエリコンピューティングコストを負担するメンバー](#)で、[クエリを行えるメンバー](#)でない場合は、を使用して予算 AWS Budgets を設定し、最大予算に達したら通知 AWS Clean Rooms を受け取ることをお勧めします。詳細については、「AWS Cost Management ユーザーガイド」の「[AWS Budgetsを用いてコストを管理する](#)」を参照してください。通知の設定の詳細については、「AWS Cost Management ユーザーガイド」の「[予算の通知に関する Amazon SNS トピックを作成する](#)」を参照してください。予算の上限に達した場合は、クエリを行えるメンバーに連絡するか、[コラボレーションから退出](#)できます。コラボレーションから退出すると、それ以上クエリの実行は許可されなくなるため、クエリの計算コストも請求されなくなります。

5. [次へ] を選択します。

コラボレーションとメンバーシップの両方が作成されます。

コラボレーションのステータスはアクティブになります。

No, I will create a membership later

1. [次へ] を選択します。

コラボレーションのみが作成されます。

コラボレーションのステータスは非アクティブになります。

9. ステップ 5: 確認して作成するには、以下を実行します。

- a. 前のステップで行った選択内容を確認し、必要に応じて編集します。
- b. いずれかのオプションを選択します。

前のステップでの選択内容	選択内容
コラボレーションと共にメンバーシップを作成 ([はい、今すぐメンバーシップを作成してご入会ください])	コラボレーションとメンバーシップを作成
コラボレーションは作成するが、現時点ではメンバーシップを作成しない ([いいえ、後でメンバーシップを作成します])	コラボレーションを作成

メンバーシップの作成とコラボレーションへの参加

メンバーシップは、AWS Clean Roomsでメンバーがコラボレーションに参加するときに作成されるリソースです。

コラボレーションには、

- [クエリを実行できるメンバー](#)
- [クエリとジョブを実行できるメンバー](#)
- [クエリまたはジョブの結果を受け取ることができるメンバー](#)

- [クエリコンピューティングコストを支払うメンバー](#)
- [クエリとジョブの支払いを行うメンバー](#)

メンバー全員がデータを寄稿できます。

AWS SDK を使用してメンバーシップを作成し、コラボレーションに参加する方法については、「[AWS Clean Rooms API リファレンス](#)」を参照してください。

この手順では、招待されたメンバーが、[メンバーシップリソースを作成してコラボレーションに参加](#)します。

招待されたメンバーが結果を受け取ることができるメンバーである場合は、結果の送信先と形式を指定します。また、結果の送信先に書き込むサービスロール ARN も提供します。

招待されたメンバーがコンピューティングコストの支払いを担当するメンバーである場合、コラボレーションに参加する前に支払い責任を受け入れます。

メンバーシップを作成してコラボレーションに参加するには

1. にサインイン AWS マネジメントコンソール し、メンバー で[AWS Clean Rooms コンソール](#)を開きます AWS アカウント。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [参加可能] タブの [参加可能なコラボレーション] で、コラボレーションの [名前] を選択します。
4. コラボレーションの詳細ページの「概要」セクションで、メンバーの詳細や他のメンバーのリストなど、コラボレーションの詳細を表示します。

コラボレーションの各メンバーの AWS アカウント IDs が、コラボレーションに入力する予定の ID であることを確認します。

5. [メンバーシップを作成] を選択します。
6. メンバーシップの作成ページの「概要」で、コラボレーション名、コラボレーションの説明、コラボレーション作成者の AWS アカウント ID、メンバーの詳細、およびクエリの支払いを行うメンバーの AWS アカウント ID を表示します。
7. コラボレーションクリエーターが分析ログ記録を有効にすることを選択した場合は、Amazon CloudWatch Logs のログストレージに次のいずれかのオプションを選択します。

選択内容	結果
オンにする	関連するログは Amazon CloudWatch Logs に保存されます。 各メンバーは、自分が開始したクエリ、または自分のデータを含むクエリのログのみを受け取ることができます。 結果を受け取ることができるメンバーは、データが分析でアクセスされていない場合でも、コラボレーションで実行されるすべての分析のログも受け取ります。 サポートされているログタイプで、コラボレーション作成者がサポート対象として選択したログタイプから選択します。 1. SQL クエリから生成されたログを受信する場合は、クエリからのログチェックボックスをオンにします。 2. PySpark を使用してジョブから生成されたログを受信する場合は、ジョブからのログチェックボックスをオンにします。
オフにする	関連するクエリログは、Amazon CloudWatch Logs アカウントに保存されません。

Note

分析ログ記録を有効にすると、ログストレージがセットアップされ、Amazon CloudWatch Logs でログの受信が開始されるまでに数分かかることがあります。この短い間、クエリを行えるメンバーがクエリを実行しても、実際にはログが送信されない場合があります。

8. コラボレーション作成者がこのコラボレーションの詳細モニタリングを有効にしている場合は、CloudWatch アカウントで詳細なオブザーバビリティメトリクスを受信するかどうかを選択します。

詳細モニタリングの場合:

詳細なモニタリングオプション

オプション	説明
オンにする	AWS Clean Rooms は、アカウント内のこのコラボレーションの詳細モニタリングメトリクスを CloudWatch に発行します。これらのメトリクスは、クエリのパフォーマンスやリソース使用率などの運用モニタリングに使用できます。 追加の CloudWatch 料金が適用されます。詳細については、「CloudWatch 料金表」をご覧ください。
オフにする	詳細なメトリクスは CloudWatch アカウントにエクスポートされません。このオプションを有効にしている場合、他のメンバーは引き続き自分のアカウントで詳細なモニタリングメトリクスを表示できます。

 Note

詳細なモニタリングメトリクスは、クエリを実行できるメンバー (分析ランナー) と、コラボレーションの支払い者として設定されたメンバーのみが使用できます。

9. メンバー機能に受信結果が含まれている場合、結果設定のデフォルトは次のとおりです。
- a. クエリ結果で、クエリのデフォルト設定を設定するチェックボックスを選択し、SAmazon S3S3 の結果の送信先を指定するか、SS3を選択して使用可能な SS3 バケットのリストから選択します。

Example

例: **s3://bucket/prefix**

- i. 結果形式には、CSV または PARQUET を選択します。

- ii. (Spark のみ) 結果ファイルの場合は、複数または単一を選択します。
- iii. (オプション) サービスアクセスの場合、S3 送信先に最大 24 時間かかるクエリを配信する場合は、最大 24 時間かかるクエリをサポートするサービスロールを追加するチェックボックスをオンにします。

完了までに最大 24 時間かかる大規模なクエリは、S3 送信先に配信されます。

チェックボックスをオンにしない場合、12 時間以内に完了したクエリのみが S3 ロケーションに配信されます。

 Note

既存のサービスロールを選択するか、新しいサービスロールを作成するアクセス許可が必要です。詳細については、「[結果を受け取るサービスロールを作成する](#)」を参照してください。

- iv. [新しいサービスロールを作成して使用] または [既存のサービスロールを使用] を選択して、[サービスアクセス] 許可を指定します。

Create and use a new service role

- AWS Clean Rooms は、このテーブルに必要なポリシーを持つサービスロールを作成します。
- デフォルトの [サービスロール名] は `cleanrooms-result-receiver-
<timestamp>` です。
- ロールを作成してポリシーをアタッチするアクセス許可が必要です。

Use an existing service role

1. ドロップダウンリストから [既存のサービスロール名] を選択します。

ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。

ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。

2. [IAM で表示] 外部リンクを選択してサービスロールを表示します。

既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。

デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

 Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、「」を参照してください [AWS の 管理ポリシー AWS Clean Rooms](#)。
- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。
- ロールポリシーを変更できない場合は、 がサービスロールのポリシーを見つけ AWS Clean Rooms られなかったことを示すエラーメッセージが表示されます。

- b. ジョブの結果で、ジョブのデフォルト設定を設定するチェックボックスを選択し、SAmazon S3S3 の結果の送信先を指定するか、SS3を選択して使用可能な SS3 バケットのリストから選択します。

Example

例: **s3://bucket/prefix**

- ドロップダウンリストから既存のサービスロール名を選択して、サービスアクセス許可を指定します。
10. メンバーシップリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
11. コラボレーションクリエイターがクエリの支払いまたはクエリとジョブの支払いを行うメンバーとしてユーザーを指定している場合は、このコラボレーションチェックボックスでコンピューティングコストの支払いに同意するを選択して、承諾を示します。

Note

続行するには、このチェックボックスをオンにする必要があります。
料金の計算方法の詳細については、「[の料金 AWS Clean Rooms](#)」を参照してください。

[クエリコンピューティングコストを支払うメンバー](#)、または[クエリとジョブコンピューティングコストを支払うメンバー](#)で、[クエリを行えるメンバー](#)でない場合は、を使用して予算 AWS Budgets を設定し、最大予算に達したら通知 AWS Clean Rooms を受け取ることをお勧めします。詳細については、「AWS Cost Management ユーザーガイド」の「[AWS Budgetsを用いてコストを管理する](#)」を参照してください。通知の設定の詳細については、「AWS Cost Management ユーザーガイド」の「[予算の通知に関する Amazon SNS トピックを作成する](#)」を参照してください。最大予算に達した場合は、クエリとジョブを実行したり、[コラボレーションを終了](#)したりできるメンバーに連絡できます。コラボレーションから退出すると、それ以上クエリの実行は許可されなくなるため、クエリの計算コストも請求されなくなります。

12. メンバーシップを作成してコラボレーションに参加してもよい場合は、[メンバーシップを作成]を選択します。

コラボレーションメタデータへの読み取りアクセスが付与されます。これには、他のメンバーのすべての名前と AWS アカウント IDs に加えて、コラボレーションの表示名や説明などの情報が含まれます。

これで次の作業に進むことができます。

- [AWS Clean Roomsでクエリ対象となるデータテーブルを準備する](#) (独自のイベントデータのクエリを実行する場合や、ID データのクエリを実行する場合は省略可能)
- [設定済みテーブルをコラボレーションに関連付ける](#) – イベントデータのクエリを実行する場合。
- [設定済みテーブルの分析ルールを追加する](#) – イベントデータのクエリを実行する場合。
- [新しい ID 名前空間を作成して関連付ける](#) – ID データをクエリする ID マッピングテーブルを作成する場合。

コラボレーションから退出する方法については、「[コラボレーションからの退出](#)」を参照してください。

コラボレーションの編集

コラボレーションクリエーターは、コラボレーションのさまざまな部分を編集できます。

AWS SDKs を使用してコラボレーションを編集する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

トピック

- [コラボレーションの名前と説明の編集](#)
- [コラボレーション分析エンジンを更新する](#)
- [ログストレージをオフにする](#)
- [コラボレーションログの設定を編集する](#)
- [コラボレーションのタグ編集](#)
- [メンバーシップのタグ編集](#)
- [新しいメンバーを追加する](#)
- [既存のメンバー機能を編集する](#)
- [コラボレーションの自動承認設定を編集する](#)
- [関連付けられているテーブルのタグ編集](#)
- [分析テンプレートのタグ編集](#)
- [差分プライバシーポリシータグの編集](#)

コラボレーションの名前と説明の編集

コラボレーションの作成後は、コラボレーションの名前と説明のみを編集できます。

コラボレーションの名前と説明を編集するには

1. にサインイン AWS マネジメントコンソール し、[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. コラボレーションの詳細ページで [アクション] を選択し、[コラボレーションを編集] を選択します。
5. コラボレーションの編集ページの 詳細 で、コラボレーションの名前と説明を編集します。

6. [Save changes] (変更の保存) をクリックします。

コラボレーション分析エンジンを更新する

コラボレーションを作成したら、分析エンジンを AWS Clean Rooms SQL から Spark に変更できます。

Note

分析エンジンを AWS Clean Rooms SQL から Spark に変更すると、既存のワークフローが破損する可能性があります。

コラボレーション分析エンジンを更新するには

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合) 。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. コラボレーションの詳細ページで [アクション] を選択し、[コラボレーションを編集] を選択します。
5. [Save changes] (変更の保存) をクリックします。

ログストレージをオフにする

分析ログ記録を有効にしている場合は、分析ログが Amazon CloudWatch Logs アカウントに保存されているかどうかを編集できます。

ログストレージをオフにするには

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合) 。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 分析ログが有効になっているコラボレーションを選択します。
4. コラボレーションの詳細ページで、アクションを選択し、ログストレージをオフにするを選択します。

Note

以下を示す警告が表示されます。

- 新しいクエリは CloudWatch アカウントに記録されなくなります。
- 既存のログは、現在の保持設定に従って保持されます。
- 今後ログ記録を再アクティブ化する場合、ログ記録は再アクティブ化後に行われたクエリにのみ適用されます。
- この変更はログにのみ影響します。他のチームメンバーのログ記録設定は変更されません。

5. [無効にする] を選択します。

コラボレーションログの設定を編集する

[クエリログ記録] を有効にしている場合は、クエリログを Amazon CloudWatch Logs アカウントに保存するかどうかを編集できます。

コラボレーションログの設定を編集するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. コラボレーションの詳細ページで、次のいずれかを実行します。
 - アクションを選択し、ログ設定の編集を選択します。
 - ログタブで、ログ設定の編集を選択します。
5. Amazon Amazon CloudWatch Logs のログストレージのログ設定の編集モーダル:
 - 関連するログを Amazon CloudWatch Logs アカウントに保存しない場合は、無効化を選択します。
 - 関連するログを Amazon CloudWatch Logs アカウントに保存する場合は、オンにします。

ログを受信できるのは、開始したクエリまたはデータを含むクエリのみです。

また、結果を受け取ることができるメンバーは、自分のデータがクエリで利用されていない場合でも、コラボレーションで実行されたすべてのクエリのログを受け取ります。

1. サポートされているログタイプで、コラボレーション作成者がサポート対象として選択したログタイプから選択します。

- SQL クエリから生成されたログを受信する場合は、クエリからのログチェックボックスをオンにします。
- PySpark を使用してジョブから生成されたログを受信する場合は、ジョブからのログチェックボックスをオンにします。

6. [Save changes] (変更の保存) をクリックします。

Note

ログ記録を有効にすると、ログストレージがセットアップされ、Amazon CloudWatch Logs でログの受信が開始されるまでに数分かかることがあります。この短い間、クエリを行えるメンバーがクエリを実行しても、実際にはログが送信されない場合があります。

コラボレーションのタグ編集

コラボレーションクリエイターは、コラボレーションの作成後に、コラボレーションリソースのタグを管理できます。

コラボレーションのタグを編集するには

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. 次のいずれかを選択します。

状況	操作
コラボレーション作成者とコラボレーションのメンバー	[詳細] タブを選択します。

状況	操作
コラボレーションクリエイターだが、コラボレーションのメンバーではない場合	ページの [タグ] セクションまで下方向にスクロールします。

5. [コラボレーションの詳細] で、[タグを管理] を選択します。
6. [Manage tags] (タグの管理) ページで、次の操作を実行できます。
 - タグを削除するには[削除] を選択してください。
 - タグを追加するには、[新しいタグの追加] を選択します。
 - 変更を保存するには、[変更内容を保存] を選択します。

メンバーシップのタグ編集

コラボレーションクリエイターは、コラボレーションの作成後に、メンバーシップリソースのタグを管理できます。

メンバーシップのタグを編集するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. [詳細] タブを選択します。
5. [メンバーシップの詳細] で [タグを管理] を選択します。
6. [メンバーシップタグを管理] ページで、次の操作を実行できます。
 - タグを削除するには[削除] を選択してください。
 - タグを追加するには、[新しいタグの追加] を選択します。
 - 変更を保存するには、変更の保存を選択します。

新しいメンバーを追加する

詳細については、「[コラボレーションに新しいメンバーを追加する](#)」を参照してください。

既存のメンバー機能を編集する

詳細については、「[既存のメンバー機能を更新する](#)」を参照してください。

コラボレーションの自動承認設定を編集する

詳細については、「[コラボレーションの自動承認設定を編集する](#)」を参照してください。

関連付けられているテーブルのタグ編集

コラボレーションクリエイターは、テーブルをコラボレーションに関連付けた後に、関連付けられているテーブルリソースのタグを管理できます。

関連付けられているテーブルのタグを編集するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. [自分が関連付けたテーブル] で、テーブルを選択します。
6. 設定済みテーブルの詳細ページの [タグ] で、[タグを管理] を選択します。

[Manage tags] (タグの管理) ページで、次の操作を実行できます。

- タグを削除するには [削除] を選択してください。
- タグを追加するには、[新しいタグの追加] を選択します。
- 変更を保存するには、変更の保存を選択します。

分析テンプレートのタグ編集

コラボレーションクリエイターは、コラボレーションの作成後に、分析テンプレートリソースのタグを管理できます。

メンバーシップのタグを編集するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。

2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. [テンプレート] タブを選択します。
5. [自分で作成した分析テンプレート] セクションで、分析テンプレートを選択します。
6. 分析テンプレートテーブルの詳細ページで、[タグ] セクションまで下方向にスクロールします。
7. [Manage tags (タグの管理)] を選択します。
8. [Manage tags] (タグの管理) ページで、次の操作を実行できます。
 - タグを削除するには[削除] を選択してください。
 - タグを追加するには、[新しいタグの追加] を選択します。
 - 変更を保存するには、変更の保存を選択します。

差分プライバシーポリシータグの編集

コラボレーションクリエイターは、コラボレーションの作成後に、分析テンプレートリソースのタグを管理できます。

メンバーシップのタグを編集するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合) 。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 編集する差分プライバシーポリシーを含むコラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. [タグ] タブで、[タグ管理] を選択します。
6. [Manage tags] (タグの管理) ページで、次の操作を実行できます。
 - タグを削除するには[削除] を選択してください。
 - タグを追加するには、[新しいタグの追加] を選択します。
 - 変更を保存するには、変更の保存を選択します。

の変更リクエストAWS Clean Rooms

変更リクエストを使用すると、他のコラボレーションメンバーによる承認のために、既存のコラボレーション設定の変更を提案できます。変更リクエストでは、新しいメンバーの追加、既存のメンバー機能の更新、コラボレーションの自動承認設定を変更するリクエストを送信できます。提案された変更を有効にするには、すべてのコラボレーションメンバーが変更リクエストを承認する必要があります。

変更リクエストはコラボレーション固有であり、コラボレーション作成者が送信できます。

変更リクエストは、次の方法で送信できます。

- コラボレーションに新しいメンバーを追加する
- 既存のメンバー機能を更新する
- コラボレーションの自動承認設定を編集する

Note

変更リクエストを送信するには、コラボレーション作成者である必要があります。

コラボレーションに新しいメンバーを追加する

コラボレーションに新しいメンバーを追加するには、コラボレーション作成者である必要があります。コラボレーションに新しいメンバーを追加するには、既存のコラボレーションメンバーからの手動承認が必要であり、変更リクエストを送信します。

以下の手順に従って、コラボレーションに新しいメンバーを追加できます。

コラボレーションに新しいメンバーを追加するには

1. にサインインAWS マネジメントコンソールし、メンバーとともに [AWS Clean Roomsコンソール](#)を開きますAWS アカウント。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択して、コラボレーションの詳細に移動します。
4. コラボレーションページで、メンバータブに移動します。

5. Members テーブルで、Edit members を選択します。
6. 別のメンバーの追加 を選択します。
7. 新しいメンバー情報を入力します。
 - メンバー表示名
 - メンバー AWS アカウントID
 - メンバーが結果を受信できるかどうかを指定します。このメンバーに権限を付与するには、チェックボックスをオンにします。
8. [Save changes] (変更の保存) をクリックします。
9. 変更リクエストの送信を確認します。確認モーダルで、変更を確認し、変更リクエストの送信を選択します。

Note

コラボレーションで自動承認の変更タイプがサポートされている場合、変更リクエストの手動承認は必要ない場合があります。変更リクエストを必要としない変更タイプは、コラボレーションの「概要」セクションで確認できます。詳細については、「[the section called “コラボレーションの自動承認設定を編集する”](#)」を参照してください。

既存のメンバー機能を更新する

既存のコラボレーションメンバー機能を更新するには、コラボレーション作成者である必要があります。既存のコラボレーションメンバーの機能を更新するには、既存のメンバーからの手動承認が必要であり、変更リクエストを送信します。

更新できるメンバー機能は次のとおりです。

- 結果を受け取ることができる

以下の手順に従って、既存のコラボレーションメンバーのメンバー機能を更新できます。

既存のメンバー機能を更新するには

1. にサインインAWS マネジメントコンソールし、メンバーとともに [AWS Clean Roomsコンソール](#)を開きますAWS アカウント。

2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択して、コラボレーションの詳細に移動します。
4. コラボレーションページで、メンバータブに移動します。
5. Members テーブルで、Edit members を選択します。
6. メンバーの変更機能を指定します。
7. [Save changes] (変更の保存) をクリックします。
8. 変更リクエストの送信を確認します。確認モーダルで、変更を確認し、変更リクエストの送信を選択します。

Note

コラボレーションで自動承認の変更タイプがサポートされている場合、変更リクエストの手動承認は必要ない場合があります。変更リクエストを必要としない変更タイプは、コラボレーションの「概要」セクションで確認できます。詳細については、「[the section called “コラボレーションの自動承認設定を編集する”](#)」を参照してください。

コラボレーションの自動承認設定を編集する

自動承認のコラボレーション設定を編集するには、コラボレーション作成者であり、他のコラボレーションメンバーによる承認のために変更リクエストを送信する必要があります。

以下の手順に従って、コラボレーションの自動承認設定を編集できます。

コラボレーションの自動承認設定を編集するには

1. にサインインAWS マネジメントコンソールし、メンバーとともに [AWS Clean Roomsコンソール](#)を開きますAWS アカウント。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択して、コラボレーションの詳細に移動します。
4. コラボレーションページで、アクションボタンを選択し、自動承認の編集を選択します。
5. 変更リクエストを手動で承認せずに既存のメンバーにメンバー機能を付与するには:
 1. メンバー権限の付与セクションに移動します。
 2. 自動的に付与できるメンバー機能を指定します。

 Note

デフォルトでは、すべてのコラボレーションメンバーは「データに貢献」できます。

3. (オプション) 変更リクエストを手動で承認せずに、新しいメンバーが指定された機能とのコラボレーションにすぐに参加できるようにするには、これらの機能を持つ新しいメンバーを自動承認を選択します。
6. 変更リクエストなしで既存のメンバーから機能を削除できるようにするには:
 1. 自動的に取り消すことができる Abilities セクションに移動します。
 2. 自動的に削除できるメンバー機能を指定します。
7. [Save changes] (変更の保存) をクリックします。
8. 変更リクエストの送信を確認します。確認モーダルで、変更を確認し、変更リクエストの送信を選択します。

コラボレーションの削除

コラボレーションクリエイターは、作成したコラボレーションを削除できます。

 Note

コラボレーションを削除すると、作成者もどのメンバーも、クエリを実行したり、結果を受け取ったり、データを寄稿したりできません。コラボレーションの各メンバーは、メンバーシップの一環として引き続き自分のデータにアクセスできます。

コラボレーションを削除するには

1. にサインイン AWS マネジメントコンソール し、 [AWS Clean Rooms コンソール](#) を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 削除するコラボレーションを選択します。
4. [アクション] で [コラボレーションを削除] を選択します。
5. 削除を確定し、[削除] を選択します。

コラボレーションの表示

コラボレーションクリエーターは、自分が作成したすべてのコラボレーションを表示できます。

コラボレーションを表示するには

1. にサインイン AWS マネジメントコンソール し、 [AWS Clean Rooms コンソール](#) を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [コラボレーション] ページの [最終使用日] に、直近に使用した 5 つのコラボレーションが表示されます。
4. [アクティブメンバーシップあり] タブには、[アクティブメンバーシップとのコラボレーション] のリストが表示されます。

[名前]、[メンバーシップの作成日]、[あなたのメンバー情報] で並べ替えることができます。

[検索] バーを使用してコラボレーションを検索できます。

5. [参加可能] タブには、[参加可能なコラボレーション] のリストが表示されます。
6. [利用できなくなりました] タブには、削除されたコラボレーションのリストと、[利用できなくなったコラボレーションのメンバーシップ] (削除されたメンバーシップ) が表示されます。

コラボレーションへのメンバーの招待

コラボレーション作成者は、コラボレーションを作成した後、メンバータブにリストされているメンバーに招待リンクを送信できます。

メンバーをコラボレーションに招待するには

1. にサインイン AWS マネジメントコンソール し、 [AWS Clean Rooms コンソール](#) を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. [メンバー] タブを選択します。
5. メンバー テーブルで、招待のコピーリンクボタンを選択します。

招待リンクがコピーされます。

6. 選択した安全な通信方法に招待リンクを貼り付け、コラボレーションの各メンバーに送信します。

メンバーのモニタリング

コラボレーションクリエイターは、コラボレーションの作成後に、[メンバー] タブですべてのメンバーのステータスを監視できます。これにより、コラボレーション全体で適切なアクセスコントロールとコスト管理を確保できます。

メンバーのステータスをモニタリングするには

1. にサインイン AWS マネジメントコンソールし、[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 作成したコラボレーションを選択します。
4. [メンバー] タブを選択します。
5. [メンバー] テーブルで、各メンバーの [ステータス] を確認します。
6. [メンバー能力] テーブルで、クエリ、結果の受信、データの寄稿、その他のタスクを実行できるメンバーを確認します。
7. [支払い設定] テーブルで、クエリ、ID マッピングテーブル、ML モデリングに対してどのメンバーが支払いを行っているかを確認します。

コラボレーションへのメンバーの追加

前提条件

- コラボレーションを管理するアクセス許可AWS アカウントを持つ
- 追加するメンバーの AWS アカウントIDs

コラボレーションにメンバーを追加するには

1. にサインインAWS マネジメントコンソールし、[AWS Clean Roomsコンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. メンバーを追加するコラボレーションを選択します。
4. [メンバー] タブを選択します。
5. [Edit members] を選択します。
6. 別のメンバーを追加を選択し、次の情報を入力します。

- メンバー表示名

- メンバー AWS アカウントID
 - 結果が受信できるかどうかを指定します。
7. [Save changes] (変更の保存) をクリックします。
 8. 確認モーダルで、情報が正しいことを確認し、変更リクエストの送信を選択します。
 - 変更リクエストに他のメンバーの承認が必要な場合、既存のメンバーはすべて、新しいメンバーを追加する前に変更リクエストを承認する必要があります。変更リクエストの詳細については、「」を参照してください[the section called “変更リクエスト”](#)。
 - 指定された機能を持つ新しいメンバーに対して自動承認変更タイプがサポートされている場合、この変更はすぐに有効になります。詳細については、「[the section called “コラボレーションの自動承認設定を編集する”](#)」を参照してください。
 9. コラボレーションの詳細ページで、メンバータブで、追加されたメンバーのメンバーステータスが招待済みとして表示されることを確認します (複数可)。

これらのステップを完了すると、招待されたメンバーはコラボレーションに参加できます。コラボレーションへの参加の詳細については、「」を参照してください。 [メンバーシップの作成とコラボレーションへの参加](#)

コラボレーションからメンバーを削除する

Note

開始する前に、メンバーを削除することに注意してください。

- コラボレーションから関連するデータセットをすべて削除します。
- [メンバーがクエリコンピューティングコストを支払う](#)と、このアクションはコラボレーション内のすべてのクエリ実行を停止します。

メンバーを削除すると、そのメンバーが関連付けられているデータセットもすべてコラボレーションから削除されます。

前提条件

- コラボレーションクリエイターである必要があります

- 自分のアカウントを削除することはできません

コラボレーションからメンバーを削除するには

1. にサインイン AWS マネジメントコンソール し、 [AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. 変更するコラボレーションを選択します。
4. [メンバー] タブを選択します。
5. 削除するメンバーの横にあるオプションボタンを選択します。
6. [削除] を選択してください。
7. 確認ダイアログボックスで、 と入力 **confirm**して削除を確認します。

 Note

メンバーを削除すると、そのアカウントに関連付けられているすべてのデータセットもコラボレーションから削除されます。

 Important

クエリコンピューティングコストを支払うメンバーを削除した場合、新しい支払いメンバーを指定するまで、コラボレーションでそれ以上クエリを実行することはできません。

コラボレーションからの退出

コラボレーションメンバーは、自分のメンバーシップを削除することでコラボレーションから退出できます。コラボレーションクリエイターは、[コラボレーションを削除](#)しないとコラボレーションから退出できません。

 Note

メンバーシップを削除すると、コラボレーションから退出することになり、再び参加することはできません。[クエリの計算コストを負担するメンバー](#)がメンバーシップを削除すると、それ以上クエリを実行できなくなります。

コラボレーションから退出するには

1. にサインイン AWS マネジメントコンソール し、 [AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [アクティブメンバーシップあり] で、自分がメンバーになっているコラボレーションを選択します。
4. [アクション] を選択します。
5. [メンバーシップを削除] を選択します。
6. ダイアログボックスで、テキスト入力フィールドに入力してコラボレーションを終了する決定を確認し、メンバーシップ **confirm** を空にして削除します。

メンバーシップが削除されたことを示すメッセージがコンソールに表示されます。

コラボレーションクリエイターには、[メンバーステータス] に [退出済み] と表示されます。

でデータテーブルを準備する AWS Clean Rooms

Note

データテーブルの準備は、コラボレーションに参加する前でも後でも行うことができます。テーブルの準備が完了した後は、そのテーブルのプライバシーニーズが同じである限り、複数のコラボレーション間でテーブルを再利用できます。

コラボレーションのメンバーは、クエリを実行できる AWS Clean Rooms コラボレーションメンバーがクエリを実行する前に、データテーブルを準備する必要があります。

のクエリに使用するデータテーブル AWS Clean Rooms は通常、他のアプリケーションで使用するのと同じタイプのデータテーブルです。例えば、Amazon Athena、Amazon EMR、Amazon Redshift Spectrum、Amazon Quick では同じタイプのデータセットが使用されます。

データは、次のいずれかのデータソースから直接元の形式でクエリできます。

- Amazon Simple Storage Service (Amazon S3)
- Amazon Athena
- Snowflake

AWS Clean Rooms はクエリの実行時にデータセットにアクセスし、クエリを実行できるメンバーが常にup-to-dateデータにアクセスできるようにします。一時的に AWS Clean Rooms コラボレーションに読み込まれたデータは、クエリの完了後に削除されます。クエリ結果は Amazon S3 バケットに書き込まれます。

ユースケースに ID データのクエリが関連している場合は、「[AWS Entity Resolution in AWS Clean Rooms](#)」を参照してください。

トピック

- [のデータ形式 AWS Clean Rooms](#)
- [Apache Iceberg の テーブル AWS Clean Rooms](#)
- [でのクエリ用のデータテーブルの準備 AWS Clean Rooms](#)
- [Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)
- [C3R 暗号化クライアントによるデータテーブルの復号化](#)

のデータ形式 AWS Clean Rooms

データを分析するには、データセットが AWS Clean Rooms サポートする形式である必要があります。

トピック

- [PySpark ジョブでサポートされているデータ形式](#)
- [SQL クエリでサポートされているデータ形式](#)
- [サポートされているデータ型](#)
- [のファイル圧縮タイプ AWS Clean Rooms](#)
- [のサーバー側の暗号化 AWS Clean Rooms](#)

PySpark ジョブでサポートされているデータ形式

AWS Clean Rooms は、PySpark ジョブを実行するために次の構造化形式をサポートしています。

- Parquet
- OpenCSV
- JSON

SQL クエリでサポートされているデータ形式

AWS Clean Rooms は、SQL クエリを実行するために次の構造化形式をサポートしています。

- [Apache Iceberg テーブル](#)
- Parquet
- OpenCSV
- JSON

Note

テキストファイル内の timestamp 値は、yyyy-MM-dd HH:mm:ss.SSSSSS の形式である必要があります。例えば、2017-05-01 11:30:59.000000 です。

Apache Parquet など、列指向ストレージファイル形式を使用することをお勧めします。列指向ストレージファイル形式を使用すると、必要な列のみを選択することで、データの移動を最小限に抑えることができます。最適なパフォーマンスを得るには、大きなオブジェクトを 100 MB ~ 1 GB のオブジェクトに分割する必要があります。

サポートされているデータ型

AWS Clean Rooms は、次のデータ型をサポートしています。

- 配列
- BIGINT
- BOOLEAN
- BYTE
- CHAR
- DATE
- DECIMAL
- FLOAT
- INTEGER
- INTERVAL
- LONG
- MAP
- REAL
- SHORT
- SMALLINT
- STRUCT
- TIME
- TIMESTAMP_LTZ
- TIMESTAMP_NTZ
- TINYINT
- VARCHAR

詳細については、AWS Clean Rooms SQL リファレンスの [「データ型」](#) を参照してください。

のファイル圧縮タイプ AWS Clean Rooms

ストレージスペースの縮小、パフォーマンスの向上、コストの最小化を行うため、データセットを圧縮することを強くお勧めします。

AWS Clean Rooms は、ファイル拡張子に基づいてファイル圧縮タイプを認識し、次の表に示す圧縮タイプと拡張子をサポートします。

圧縮アルゴリズム	ファイル拡張子
GZIP	.gz
Bzip2	.bz2
Snappy	.snappy

さまざまなレベルで圧縮を適用できます。通常、ファイル全体を圧縮するか、ファイル内の個々のブロックを圧縮します。ファイルレベルで列形式を圧縮しても、パフォーマンス上の利点はありません。

のサーバー側の暗号化 AWS Clean Rooms

Note

暗号化コンピューティングを必要とするユースケースで、サーバー側の暗号化が代替の役割を果たすことはありません。

AWS Clean Rooms は、次の暗号化オプションを使用して暗号化されたデータセットを透過的に復号します。

- SSE-S3 – Amazon S3 によって管理される AES-256 暗号化キーを使用したサーバー側暗号化。
- SSE-KMS – によって管理されるキーによるサーバー側の暗号化 AWS Key Management Service

SSE-S3 を使用するには、設定済みテーブルをコラボレーションに関連付けるために使用される AWS Clean Rooms サービスロールに KMS 復号アクセス許可が必要です。SSE-KMS を使用するには、KMS キーポリシーで AWS Clean Rooms サービスロールの復号も許可する必要があります。

AWS Clean Rooms は Amazon S3 クライアント側の暗号化をサポートしていません。サーバー側の暗号化の詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[サーバー側の暗号化によるデータの保護](#)」を参照してください。

Apache Iceberg の テーブル AWS Clean Rooms

Apache Iceberg は、データレイクのオープンソーステーブル形式です。は、Apache Iceberg メタデータに保存されている統計を使用してクエリプランを最適化し、クリーンルームのクエリ処理中のファイルスキャンを減らす AWS Clean Rooms ことができます。詳細については、[Apache Iceberg](#) のドキュメントを参照してください。

Iceberg テーブル AWS Clean Rooms で を使用する場合は、次の点を考慮してください。

- S3 用の Apache Iceberg テーブル – Apache Iceberg テーブルは、オープンソースのグルーカタログの実装 AWS Glue Data Catalog に基づいて で定義する必要があります。 <https://iceberg.apache.org/docs/latest/aws/#glue-catalog>
- Athena の Apache Iceberg テーブル – 詳細については、 <https://docs.aws.amazon.com/athena/latest/ug/querying-iceberg.html> を参照してください。
- Snowflake 用の Apache Iceberg テーブル – 詳細については、 <https://docs.snowflake.com/en/user-guide/tables-iceberg> を参照してください。
- Parquet ファイル形式 – は、Parquet データファイル形式の Iceberg テーブル AWS Clean Rooms のみをサポートします。
- GZIP および Snappy 圧縮 – GZIP および Snappy 圧縮を備えた Parquet AWS Clean Rooms をサポートします。
- Iceberg バージョン – バージョン 1 およびバージョン 2 の Iceberg テーブルに対するクエリの実行 AWS Clean Rooms をサポートします。
- パーティション – Apache Iceberg テーブルのパーティションを手動で追加する必要はありません AWS Glue。 は Apache Iceberg テーブル内の新しいパーティションを自動的に AWS Clean Rooms 検出し、テーブル定義のパーティションを更新するために手動操作は必要ありません。Iceberg パーティションは、AWS Clean Rooms テーブルスキーマでは通常の列として表示され、設定済みテーブルスキーマではパーティションキーとして個別に表示されません。
- 制限
 - 新しい Iceberg テーブルのみ

Apache Iceberg テーブルから変換された Apache Parquet テーブルはサポートされていません。

- タイムトラベルクエリ

AWS Clean Rooms は、Apache Icebergテーブルでのタイムトラベルクエリをサポートしていません。

- Athena エンジンバージョン 2

Athena エンジンバージョン 2 で作成されたIceberg テーブルはサポートされていません。

- ファイル形式

Avro および最適化された行列 (ORC) のファイル形式はサポートされていません。

- 圧縮

Parquet の Zstandard (Zstd) 圧縮はサポートされていません。

Iceberg テーブルでサポートされているデータ型

AWS Clean Rooms は、次のデータ型を含むIcebergテーブルをクエリできます。

- BOOLEAN
- DATE
- DECIMAL
- DOUBLE
- FLOAT
- INT
- LIST
- LONG
- MAP
- STRING
- STRUCT
- TIMESTAMP WITHOUT TIME ZONE

Iceberg のデータ型の詳細については、Apache Iceberg ドキュメントで [Iceberg のスキーマ](#)を参照してください。

でのクエリ用のデータテーブルの準備 AWS Clean Rooms

ユースケースで独自のデータを持ち込む必要がない場合は、この手順をスキップできます。

ユースケースに ID データのクエリが関連している場合は、「[AWS Entity Resolution in AWS Clean Rooms](#)」を参照してください。

使用できるデータ形式の詳細については、「」を参照してください [のデータ形式 AWS Clean Rooms](#)。

トピック

- [Amazon S3 でのデータテーブルの準備](#)
- [Amazon Athena でのデータテーブルの準備](#)
- [Snowflake でのデータテーブルの準備](#)

Amazon S3 でのデータテーブルの準備

でカタログ化 AWS Glue され、Amazon S3 に保存されているデータテーブルを分析できます。データテーブルがすでにカタログ化されている場合は AWS Glue、「」に進みます [で設定済みテーブルを作成する AWS Clean Rooms](#)。

Important

で使用するデータを Amazon S3 で準備するときは AWS Clean Rooms、AWS Glue テーブルの場所がデータファイルが保存されているディレクトリパスと正確に一致することを確認する必要があります。

例: データが `s3://mybucket/folder/subfolder/data.parquet`` に保存されている場合、AWS Glue テーブルの場所は ` を指す必要があります `s3://mybucket/folder/subfolder/`。テーブルの場所を親ディレクトリ (`s3://mybucket/folder/``) に設定すると、クエリを実行するとテーブルが空になります。

Amazon S3 でデータテーブルを準備するには、次のステップを実行します。

トピック

- [ステップ 1: 前提条件を満たす](#)
- [ステップ 2: \(オプション\) 暗号化コンピューティング用のデータを準備する](#)

- [ステップ 3: データテーブルを Amazon S3 にアップロードする](#)
- [ステップ 4: AWS Glue テーブルを作成する](#)
- [ステップ 5: 次のステップ](#)

ステップ 1: 前提条件を満たす

で使用するデータテーブルを準備するには AWS Clean Rooms、次の前提条件を満たす必要があります。

- データテーブルは、[サポートされているデータ形式の AWS Clean Rooms](#) 1 つとして保存されます。
- データテーブルは にカタログ化 AWS Glue され、[サポートされているデータ型 AWS Clean Rooms](#)を使用します。
- すべてのデータテーブルは、コラボレーションが作成されたのと同じ AWS リージョン の Amazon Simple Storage Service (Amazon S3) に保存されます。
- は、コラボレーションと同じリージョンにある AWS Glue Data Catalog 必要があります。
- AWS Glue Data Catalog はメンバーシップ AWS アカウント と同じ にあります。
- Amazon S3 バケットは に登録されていません AWS Lake Formation。

ステップ 2: (オプション) 暗号化コンピューティング用のデータを準備する

(オプション) 暗号化コンピューティングを使用していて、データテーブルに暗号化が必要な機密情報が含まれている場合は、C3R 暗号化クライアントを使用してデータテーブルを暗号化する必要があります。

暗号化コンピューティング用のデータを準備するには、「[Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)」の手順に従ってください。

ステップ 3: データテーブルを Amazon S3 にアップロードする

Note

暗号化されたデータテーブルをコラボレーションで使用する場合は、データテーブルを Amazon S3 にアップロードする前に、まずデータを暗号化コンピューティング用に暗号化する必要があります。詳細については、「[Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)」を参照してください。

データテーブルを Amazon S3 にアップロードするには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。
2. [バケット] を選択し、データテーブルを保存するバケットを選択します。
3. [アップロード] を選択し、プロンプトに従います。
4. [オブジェクト] タブを選択し、データが保存されているプレフィックスを表示します。フォルダの名前を書き留めます。

フォルダを選択すると、データが表示されます。

ステップ 4: AWS Glue テーブルを作成する

データ AWS Glue テーブルが既にある場合は、このステップをスキップできます。

このステップでは、S3 バケット内のすべてのファイルをクロールし、AWS Glue AWS Glue テーブルを作成するクローラを にセットアップします。詳細については、「AWS Glue ユーザーガイド」の「[AWS Glueでのクローラの定義](#)」を参照してください。

サポートされている AWS Glue Data Catalog データ型の詳細については、「」を参照してください [サポートされているデータ型](#)。

Note

AWS Clean Rooms は現在、 に登録されている S3 バケットをサポートしていません AWS Lake Formation。

次の手順では、AWS Glue テーブルを作成する方法について説明します。AWS Key Management Service (AWS KMS) キーで暗号化された AWS Glue Data Catalog オブジェクトを使用する場合は、その暗号化されたテーブルへのアクセスを許可するように KMS キーのアクセス許可ポリシーを設定する必要があります。詳細については、「AWS Glue デベロッパーガイド」の「[AWS Glue での暗号化のセットアップ](#)」を参照してください。

AWS Glue テーブルを作成するには

1. 「AWS Glue ユーザーガイド」の「[AWS Glue コンソールでのクローラの使用](#)」の手順に従います。
2. AWS Glue データベース名と AWS Glue テーブル名を書き留めます。

ステップ 5: 次のステップ

Amazon S3 でデータテーブルを準備したので、次の準備が整いました。

- [設定済みテーブルを作成する](#)
- [ML モデルを作成する](#)

テーブルは、次の後にクエリできます。

- コラボレーションクリエイターが AWS Clean Rooms でコラボレーションを設定していること。詳細については、「[コラボレーションの作成](#)」を参照してください。
- コラボレーションクリエイターから、コラボレーションの参加者としてコラボレーション ID が送信されていること。

Amazon Athena でのデータテーブルの準備

Amazon Athena で AWS Glue Data Catalog (GDC) ビューとして作成されたデータテーブルをクエリできます。

GDC ビューは、1 つ以上の基盤となるテーブルから作成された仮想 AWS Glue テーブルです。Athena AwsGlueCatalog カタログの Athena SQL を使用して作成する必要があります。

Amazon Athena でデータテーブルを準備するには、次のステップを実行します。

トピック

- [ステップ 1: 前提条件を満たす](#)
- [ステップ 2: \(オプション\) 暗号化コンピューティング用のデータを準備する](#)
- [ステップ 3: 次のステップ](#)

ステップ 1: 前提条件を満たす

で使用するデータテーブルを準備するには AWS Clean Rooms、次の前提条件を満たす必要があります。

- データテーブルは、[サポートされているデータ形式の AWS Clean Rooms](#) 1 つとして保存されます。
- データテーブルは、[サポートされているデータ型 AWS Clean Rooms](#)を使用します。

- Athena AwsDataCatalogカタログの Athena SQL を使用して AWS Glue テーブルに GDC ビューを作成しました。

ビューは次の場所に表示されます。

- ビューとしての Athena コンソール (の下AwsDataCatalog): <https://console.aws.amazon.com/athena/>
- AWS Glue テーブルとしての AWS Glue コンソール: <https://console.aws.amazon.com/glue/>

詳細については、Amazon Athena [Athena ユーザーガイド](#) の「[Athena でデータカタログビューを使用する](#)」を参照してください。

Note

Athena と でビューを作成するには、適切なアクセス許可が必要です AWS Glue。また、ビュー定義で参照されている基盤となるテーブルにアクセスできることを確認してください。

AWS Clean Rooms は、Lambda または Hive AWS Glue カタログタイプではなく、Athena のカタログタイプのみをサポートします。

- データテーブルまたは GDC ビューは にカタログ化 AWS Glue され、 に登録されます AWS Lake Formation。
- Athena の結果を受け取るために、Amazon S3 に別の出力バケットを作成しました。
- Amazon Athena からデータを読み取るようにサービスロールを設定しました。詳細については、「[Amazon Athena からデータを読み取るサービスロールを作成する](#)」を参照してください。
- サービスロールには、GDC ビューまたはテーブルに対する Lake Formation Select および Describe アクセス許可があります。

ステップ 2: (オプション) 暗号化コンピューティング用のデータを準備する

(オプション) 暗号化コンピューティングを使用していて、データテーブルに暗号化が必要な機密情報が含まれている場合は、C3R 暗号化クライアントを使用してデータテーブルを暗号化する必要があります。

暗号化コンピューティング用のデータを準備するには、「[Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)」の手順に従ってください。

ステップ 3: 次のステップ

Amazon Athena でデータテーブルを準備したら、次の準備が整いました。

- [設定済みテーブルを作成する](#)
- [ML モデルを作成する](#)

テーブルは、次の後にクエリできます。

- コラボレーションクリエイターが AWS Clean Rooms でコラボレーションを設定していること。詳細については、「[コラボレーションの作成](#)」を参照してください。
- コラボレーションクリエイターから、コラボレーションの参加者としてコラボレーション ID が送信されていること。

Snowflake でのデータテーブルの準備

Snowflake データウェアハウスに保存されているデータテーブルをクエリできます。

Snowflake でデータテーブルを準備するには、次のステップを実行します。

トピック

- [ステップ 1: 前提条件を満たす](#)
- [ステップ 2: \(オプション\) 暗号化コンピューティング用のデータを準備する](#)
- [ステップ 3: AWS Secrets Manager シークレットを作成する](#)
- [ステップ 4: 次のステップ](#)

ステップ 1: 前提条件を満たす

で使用するデータテーブルを準備するには AWS Clean Rooms、次の前提条件を満たす必要があります。

- データテーブルを読み取るための適切なアクセス許可 AWS アカウント が付与された があります。詳細については、「[Snowflake からデータを読み取るサービスロールを作成する](#)」を参照してください。
- データテーブルは、[サポートされているデータ形式の AWS Clean Rooms](#) 1 つとして保存されます。

- データテーブルは、[サポートされているデータ型 AWS Clean Rooms](#)を使用します。
- データテーブルは Snowflake ウェアハウスに保存されます。詳細については、[Snowflake ドキュメント](#)を参照してください。
- コラボレーションに関連付ける Snowflake テーブルへの読み取り専用権限を持つ新しい Snowflake ユーザーをセットアップしました。

ステップ 2: (オプション) 暗号化コンピューティング用のデータを準備する

(オプション) 暗号化コンピューティングを使用していて、データテーブルに暗号化が必要な機密情報が含まれている場合は、C3R 暗号化クライアントを使用してデータテーブルを暗号化する必要があります。

暗号化コンピューティング用のデータを準備するには、「[Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備](#)」の手順に従ってください。

ステップ 3: AWS Secrets Manager シークレットを作成する

から Snowflake に接続するには AWS Clean Rooms、Snowflake 認証情報を作成して AWS Secrets Manager シークレットに保存し、そのシークレットを Snowflake テーブルに関連付ける必要があります AWS Clean Rooms。

Note

専用の新しいユーザーを作成することをお勧めします AWS Clean Rooms。そのユーザーには、AWS Clean Rooms アクセスするデータの読み取り権限を持つロールのみが必要です。

AWS Secrets Manager シークレットを作成するには

1. Snowflake でユーザーを生成snowflakeUserし、キーペア認証を設定します。

Note

2025 年 11 月、Snowflake はキーペア認証のみのサポートに移行します。この変更は、ユーザー名とパスワード認証を使用する Snowflake との現在の AWS Clean Rooms 統合に影響します。この日以降、の Snowflake 接続 AWS Clean Rooms では、Snowflake Privacy Enhanced Mail (PEM) プライベートキーを使用したキーペア認証が必要になります。

2. このユーザーが操作する Snowflake ウェアハウス を決定します snowflakeWarehouse。Snowflake snowflakeUser で DEFAULT_WAREHOUSE の として設定するか、次のステップで覚えておいてください。
3. [AWS Secrets Manager](#) で、Snowflake の認証情報を使用してシークレットを作成します。Secrets Manager でシークレットを作成するには、AWS Secrets Manager 「ユーザーガイド」の「[シークレットの作成](#)」にあるチュートリアルに従います。シークレットを作成したら、次のステップ secretName でシークレット名を保持します。
 - キーと値のペアを選択するときは、キー snowflakeUser を使用して のペアを作成します sfUser。
 - キーと値のペアを選択するときは、Snowflake PEM プライベートキーのペアをキー で作成します pem_private_key。
 - キーと値のペアを選択するときは、キー snowflakeWarehouse を使用して のペアを作成します sfWarehouse。

これは、Snowflake でデフォルトが設定されている場合は必要ありません。

- キーと値のペアを選択するときは、キー snowflakeRole を使用して のペアを作成します sfRole。

ステップ 4: 次のステップ

Snowflake でデータテーブルを準備したら、次の準備が整いました。

- [設定済みテーブルを作成する](#)
- [ML モデルを作成する](#)

テーブルは、次の後にクエリできます。

- コラボレーションクリエイターが AWS Clean Rooms でコラボレーションを設定していること。詳細については、「[コラボレーションの作成](#)」を参照してください。
- コラボレーションクリエイターから、コラボレーションの参加者としてコラボレーション ID が送信されていること。

Cryptographic Computing for Clean Rooms による暗号化データテーブルの準備

Cryptographic Computing for Clean Rooms (C3R) は の機能です AWS Clean Rooms。C3R を使用して、任意の当事者やコラボレーション AWS で AWS Clean Rooms 学習できる内容を暗号的に制限できます。

データテーブルをデータソースにアップロードする前に、クライアント側の暗号化ツールである C3R 暗号化クライアントを使用してデータテーブルを暗号化できます。Amazon Simple Storage Service (Amazon S3)、Amazon Athena、または Snowflake。

詳細については、「[Cryptographic Computing for Clean Rooms](#)」を参照してください。

C3R で暗号化されたデータテーブルを準備するには、以下のステップに従います。

Steps

- [ステップ 1: 前提条件を満たす](#)
- [ステップ 2: C3R 暗号化クライアントをダウンロードする](#)
- [ステップ 3: \(オプション\) C3R 暗号化クライアントで使用可能なコマンドを表示する](#)
- [ステップ 4: 表形式ファイルの暗号化スキーマを生成する](#)
- [ステップ 5: 共有シークレットキーを作成する](#)
- [ステップ 6: 共有シークレットキーを環境変数に保存する](#)
- [ステップ 7: データを暗号化する](#)
- [ステップ 8: データ暗号化を確認する](#)
- [\(オプション\) スキーマの作成 \(上級ユーザー\)](#)

ステップ 1: 前提条件を満たす

C3R で使用するデータテーブルを準備するには、以下の前提条件を満たす必要があります。

- GitHub の Cryptographic Computing for Clean Rooms リポジトリにアクセスできること。

<https://github.com/aws/c3r>

- C3R 暗号化クライアントを使用するように AWS 認証情報を設定しました。これらの認証情報は、C3R 暗号化クライアントがコラボレーションメタデータを取得 AWS Clean Rooms するた

めのへの読み取り専用 API コールに使用されます。詳細については、「AWS Command Line Interface バージョン 2 ユーザーガイド」の「[AWS CLIを設定する](#)」を参照してください。

- 使用中のマシンに Java Runtime Environment (JRE) 11 以降がインストールされていること。
 - 推奨される Java Runtime Environment、Amazon Corretto 11 以降は、<https://aws.amazon.com/corretto> からダウンロードできます。
 - Java Development Kit (JDK) には、同じバージョンの対応する JRE が含まれています。ただし、Cryptographic Computing for Clean Rooms (C3R) 暗号化クライアントを実行するために、JDK の追加機能は必要ありません。
- 表形式のデータファイル (.csv) または Parquet ファイル (.parquet) がローカルに保存されていること。
- 自分またはコラボレーションに参加している他のメンバーが、共有シークレットキーを作成できること。詳細については、「[ステップ 5: 共有シークレットキーを作成する](#)」を参照してください。
- コラボレーションクリエーターは、コラボレーションが有効になっている暗号化コンピューティング AWS Clean Rooms を使用して、コラボレーションを作成しました。詳細については、「[コラボレーションの作成](#)」を参照してください。
- コラボレーションクリエーターから、コラボレーションの参加者としてコラボレーション ID が送信されていること。送信された招待状にはコラボレーションの Amazon リソースネーム (ARN) が含まれ、コラボレーション ID が示されています。

ステップ 2: C3R 暗号化クライアントをダウンロードする

GitHub から C3R 暗号化クライアントをダウンロードするには

1. Cryptographic Computing for Clean Rooms AWS GitHubリポジトリに移動します: <https://github.com/aws/c3r>
2. ファイルを選択してダウンロードします。

ソースコード、ライセンス、および関連資料は、GitHub リポジトリのランディングページから複製、または zip ファイルとしてダウンロードできます (リポジトリのコンテンツリストの右上にある [Code] ボタンを参照してください)。

最新の署名済み C3R 暗号化クライアントの Java Executable File (つまり、コマンドラインインターフェイスアプリケーション) は、GitHub リポジトリの [Releases] ページにあります。

Apache Spark 用の C3R 暗号化クライアントパッケージ (c3r-cli-spark) は c3r-cli の 1 バージョンであり、実行中の Apache Spark サーバーにジョブとして送信する必要があります。詳細については、「[Running C3R on Apache Spark](#)」を参照してください。

ステップ 3: (オプション) C3R 暗号化クライアントで使用可能なコマンドを表示する

以下の手順に従って、C3R 暗号化クライアントで使用可能なコマンドを理解してください。

C3R 暗号化クライアントで使用可能なコマンドをすべて表示するには

1. コマンドラインインターフェイス (CLI) から、ダウンロードした c3r-cli.jar ファイルが含まれているフォルダに移動します。
2. 次のコマンドを実行します。 `java -jar c3r-cli.jar`
3. 使用可能なコマンドとオプションが一覧表示されます。

ステップ 4: 表形式ファイルの暗号化スキーマを生成する

データを暗号化するには、データの使用方法を記述した暗号化スキーマが必要です。このセクションでは、ヘッダー行を含む CSV ファイルまたは Parquet ファイルの暗号化スキーマが、C3R 暗号化クライアントによってどのように生成されるかについて説明します。

この操作はファイルごとに 1 回のみ必要です。スキーマを作成したら、それを再利用して同じファイル (または同じ列名を持つ任意のファイル) を暗号化できます。列名または必要な暗号化スキーマが変更された場合は、スキーマファイルを更新する必要があります。詳細については、「[\(オプション\) スキーマの作成 \(上級ユーザー\)](#)」を参照してください。

Important

コラボレーションを行うすべての関係者が同じ共有シークレットキーを使用することが最も重要です。また、コラボレーションの関係者は、クエリで列を JOIN する場合や等価比較を行う場合に、列名が一致するように調整する必要があります。これを行わないと、SQL クエリが予期しない結果や誤った結果を生成する可能性があります。ただし、コラボレーションクリエイターがコラボレーションの作成中に `allowJoinsOnColumnsWithDifferentNames` 暗号化設定を有効にしている場合は、これ

は必要ありません。暗号化に関する設定の詳細については、「[暗号コンピューティングパラメータ](#)」を参照してください。

スキーマモードで実行すると、C3R 暗号化クライアントは入力ファイルを列ごとにチェックし、その列を処理すべきかどうか、またどのように処理すべきかのプロンプトを表示します。ファイルに暗号化後の出力では必要のない列が多数含まれていると、不要な列をそれぞれスキップしなければならないため、スキーマ生成のインタラクティブなプロセスが面倒になることがあります。これを回避するには、スキーマを手動で記述するか、必要な列のみを含む簡略版の入力ファイルを作成します。そうすることで、その縮小されたファイルに対してインタラクティブなスキーマ生成プロセスを実行できます。C3R 暗号化クライアントはスキーマファイルに関する情報を出力し、ソース列をターゲット出力にどのように含めるか、またはどのように暗号化するか (暗号化する場合) をたずねます。

入力ファイルのソース列ごとに、次の入力を求められます。

1. 生成するターゲット列の数
2. 各ターゲット列を暗号化する方法 (暗号化する場合)
3. 各ターゲット列の名前
4. 列をsealed列として暗号化する場合は、暗号化前にデータをどのようにパディングするか

 Note

sealed列として暗号化されている列のデータを暗号化する場合は、どのデータをパディングする必要があるかを判断する必要があります。C3R 暗号化クライアントでは、スキーマ生成時に、列のすべてのエントリを同じ長さにするデフォルトのパディングを推奨します。fixed の長さを決定する際、パディングはビット単位ではなくバイト単位であることに注意してください。

以下はスキーマを作成するための決定表です。

スキーマの決定表

決定	Number of target columns from source column <'name-of-column'> ?	Target column type: [c] cleartext, [f] fingerprint, or [s] sealed ?	Target column headername <default 'name-of-column'>	Add suffix <suffix> to header to indicate how it was encrypted, [y] yes or [n] no <default 'yes'>	<'name-of-column_sealed'> padding type: [n] one, [f] fixed, or [m] max <default 'max'>
列を暗号化しない。	1	c	該当しない	該当しない	該当しない
列をfingerprint列として暗号化する。	1	f	デフォルトを選択するか、新しいヘッダ一名を入力します。	「y」を入力してデフォルト (_fingerprint) を選択するか、「n」を入力します。	該当しない
列をsealed列として暗号化する。	1	s	デフォルトを選択するか、新しいヘッダ一名を入力します。	「y」を入力してデフォルト (_sealed) を選択するか、「n」を入力します。	パディングタイプを選択します。 詳細については、「 (オプション) スキーマの作成 (上級ユーザー) 」を参照してください。

決定	Number of target columns from source column <'name-of-column'> ?	Target column type: [c] cleartext, [f] fingerprint, or [s] sealed ?	Target column headername <default 'name-of-column'>	Add suffix <suffix> to header to indicate how it was encrypted, [y] yes or [n] no <default 'yes'>	<'name-of-column_sealed'> padding type: [n] one, [f] fixed, or [m] max <default 'max'>
列をfingerprintとsealedの両方として暗号化する。	2	1 番目のターゲット列に「f」と入力します。 2 番目のターゲット列に「s」と入力します。	各ターゲット列のターゲットヘッダーを選択します。	「y」を入力してデフォルトを選択するか、「n」を入力します。	パディングタイプを選択します (sealed列のみ)。 詳細については、「 (オプション) スキーマの作成 (上級ユーザー) 」を参照してください。

次の 2 つの例は、暗号化スキーマの作成方法を示しています。インタラクションの正確な内容は、入力ファイルとユーザーが入力した回答によって異なります。

例

- [例: fingerprint列とcleartext列の暗号化スキーマの生成](#)
- [例:sealed、fingerprint、およびcleartext列を含む暗号化スキーマの生成](#)

例: fingerprint列とcleartext列の暗号化スキーマの生成

この例では、ads.csv に、username と ad_variant の 2 つの列しかありません。これらの列を次のようにします。

- username 列を fingerprint 列として暗号化する
- ad_variant 列を cleartext 列にする

fingerprint列とcleartext列の暗号化スキーマを生成するには

1. (オプション) c3r-cli.jar ファイルと暗号化するファイルが存在することを確認します。
 - a. 目的のディレクトリに移動し、ls (Mac または Unix/Linux を使用している場合)、あるいは dir (Windows を使用している場合) を実行します。
 - b. 表形式のデータファイル (.csv など) のリストを表示し、暗号化するファイルを選択します。

この例では、ads.csv が暗号化するファイルです。

2. CLI から、次のコマンドを実行してスキーマをインタラクティブに作成します。

```
java -jar c3r-cli.jar schema ads.csv --interactive --output=ads.json
```

Note

- `java --jar PATH/T0/c3r-cli.jar` を実行することができます。または、CLASSPATH 環境変数に `PATH/T0/c3r-cli.jar` を追加した場合は、クラス名を実行することもできます。C3R 暗号化クライアントは CLASSPATH をチェックしてクラス名を検索します (`java com.amazon.psion.cli.Main` など)。
- `--interactive` フラグは、インタラクティブモードでのスキーマ作成を選択するものです。これにより、ユーザーはウィザードに従ってスキーマを作成することになります。高度なスキルを持つユーザーは、ウィザードを使用せずに独自のスキーマ JSON を作成できます。詳細については、「[\(オプション\) スキーマの作成 \(上級ユーザー\)](#)」を参照してください。
- `--output` フラグは出力名を設定するものです。--output フラグを含めないと、C3R 暗号化クライアントはデフォルトの出力名 (`<input>.out.csv`、またはスキーマには `<input>.json` など) を選択しようとします。

3. Number of target columns from source column 'username'? で「1」を入力して、Enter キーを押します。
4. Target column type: [c]leartext, [f]ingerprint, or [s]ealed? で「f」を入力して、Enter キーを押します。

5. Target column headername <default 'username'> で Enter キーを押します。

デフォルトのユーザー名「username」が使用されます。

6. Add suffix '_fingerprint' to header to indicate how it was encrypted, [y]es or [n]o <default 'yes'> で「y」を入力して、Enter キーを押します。

Note

インタラクティブモードでは、暗号化された列のヘッダーに追加するサフィックス (fingerprint列の場合は _fingerprint、sealed列の場合は _sealed) が提示されます。サフィックスは、へのデータのアップロード AWS のサービス や AWS Clean Rooms コラボレーションの作成などのタスクを実行する場合に役立ちます。これらのサフィックスによって、各列の暗号化されたデータで何ができるかを判別できます。例えば、sealed列 (_sealed) として暗号化した列に JOIN を実行しようとしても (またはその逆を試みても)、正常に処理されません。

7. Number of target columns from source column 'ad_variant'? で「1」を入力して、Enter キーを押します。
8. Target column type: [c]leartext, [f]ingerprint, or [s]ealed? で「c」を入力して、Enter キーを押します。
9. Target column headername <default 'username'> で Enter キーを押します。

デフォルトのユーザー名「ad_variant」が使用されます。

スキーマが ads.json という新しいファイルに書き込まれます。

Note

スキーマは Windows の Notepad や macOS の TextEdit などの任意のテキストエディタで開いて表示できます。

10. これで、[データを暗号化](#)する準備ができました。

例:sealed、fingerprint、およびcleartext列を含む暗号化スキーマの生成

この例では、sales.csv に、username、purchased、および product の 3 つの列があります。これらの列を次のようにします。

- product 列を sealed 列にする
- username 列を fingerprint 列として暗号化する
- purchased 列を cleartext 列にする

sealed、fingerprint、およびcleartext列を含む暗号化スキーマを生成するには

1. (オプション) c3r-cli.jar ファイルと暗号化するファイルが存在することを確認します。
 - a. 目的のディレクトリに移動し、ls (Mac または Unix/Linux を使用している場合)、あるいは dir (Windows を使用している場合) を実行します。
 - b. 表形式のデータファイル (.csv) のリストを表示し、暗号化するファイルを選択します。

この例では、sales.csv が暗号化するファイルです。

2. CLI から、次のコマンドを実行してスキーマをインタラクティブに作成します。

```
java -jar c3r-cli.jar schema sales.csv --interactive --  
output=sales.json
```

Note

- --interactive フラグは、インタラクティブモードでのスキーマ作成を選択するものです。これにより、ユーザーはガイド付きワークフローに従ってスキーマを作成することになります。
- 高度なスキルを持つユーザーは、ガイド付きのワークフローを使用せずに独自のスキーマ JSON を作成できます。詳細については、「[\(オプション\) スキーマの作成 \(上級ユーザー\)](#)」を参照してください。
- 列ヘッダーのない .csv ファイルについては、CLI で使用可能なスキーマコマンドの --noHeaders フラグを参照してください。
- --output フラグは出力名を設定するものです。--output フラグを含めないと、C3R 暗号化クライアントはデフォルトの出力名 (<input>.out、またはスキーマには <input>.json など) を選択しようとします。

3. Number of target columns from source column 'username'? で「1」を入力して、Enter キーを押します。
4. Target column type: [c]leartext, [f]ingerprint, or [s]ealed? で「f」を入力して、Enter キーを押します。

5. Target column headername <default 'username'> で Enter キーを押します。

デフォルトのユーザー名「username」が使用されます。

6. Add suffix '_fingerprint' to header to indicate how it was encrypted, [y]es or [n]o <default 'yes'> で「y」を入力して、Enter キーを押します。
7. Number of target columns from source column 'purchased'? で「1」を入力して、Enter キーを押します。
8. Target column type: [c]leartext, [f]ingerprint, or [s]ealed? で「c」を入力して、Enter キーを押します。
9. Target column headername <default 'purchased'> で Enter キーを押します。

デフォルトのユーザー名「purchased」が使用されます。

10. Number of target columns from source column 'product'? で「1」を入力して、Enter キーを押します。
11. Target column type: [c]leartext, [f]ingerprint, or [s]ealed? で「s」を入力して、Enter キーを押します。
12. Target column headername <default 'product'> で Enter キーを押します。

デフォルトのユーザー名「product」が使用されます。

13. 'product_sealed' padding type: [n]one, [f]ixed, or [m]ax <default 'max'?> で Enter キーを押してデフォルトを選択します。
14. Byte-length beyond max length to pad cleartext to in 'product_sealed' <default '0'?> で Enter キーを押してデフォルトを選択します。

スキーマが sales.json という新しいファイルに書き込まれます。

15. これで、[データを暗号化](#)する準備ができました。

ステップ 5: 共有シークレットキーを作成する

データテーブルを暗号化するには、コラボレーション参加者が共有シークレットキーについて合意し、キーを安全に共有する必要があります。

共有シークレットキーは 256 ビット (32 バイト) 以上である必要があります。これより大きいキーを指定することもできますが、それによってセキュリティがさらに強化されることはありません。

⚠ Important

暗号化と復号化に使用されるキーとコラボレーション ID は、すべてのコラボレーション参加者で同一でなければならないことに注意してください。

以下のセクションでは、各端末の現在の作業ディレクトリに `secret.key` として保存される共有シークレットキーを生成するコンソールコマンドの例を示します。

トピック

- [例: OpenSSL を使用したキーの生成](#)
- [例: Windows の PowerShell を使用したキーの生成](#)

例: OpenSSL を使用したキーの生成

一般的な汎用目的の暗号化ライブラリでは、以下のコマンドを実行して共有シークレットキーを作成します。

```
openssl rand 32 > secret.key
```

Windows を使用していてまだ OpenSSL をインストールしていない場合は、「[例:Windows の PowerShell を使用したキーの生成](#)」で説明されている例を使用してキーを生成できます。

例: Windows の PowerShell を使用したキーの生成

Windows で利用可能なターミナルアプリケーションの PowerShell では、以下のコマンドを実行して共有シークレットキーを作成します。

```
$bs = New-Object Byte[](32);  
[Security.Cryptography.RandomNumberGenerator]::Create().GetBytes($bs); Set-  
Content 'secret.key' -Encoding Byte -Value $bs
```

ステップ 6: 共有シークレットキーを環境変数に保存する

環境変数は、ユーザーがなどのさまざまなキーストアからシークレットキーを提供 AWS Secrets Manager して C3R 暗号化クライアントに渡すための便利で拡張可能な方法です。

を使用してそれらのキー AWS CLI を関連する環境変数に保存 AWS のサービス する場合、C3R 暗号化クライアントは に保存されているキーを使用できます。たとえば、C3R 暗号化クライアントは

のキーを使用できます AWS Secrets Manager。詳細については、「[AWS Secrets Manager ユーザーガイド](#)」の「[AWS Secrets Managerでのシークレットの作成と管理](#)」を参照してください。

Note

ただし、AWS のサービス などの を使用して C3R キー AWS Secrets Manager を保持する前に、ユースケースで許可されていることを確認してください。特定のユースケースでは、キーの保留が必要になる場合があります AWS。これは、暗号化されたデータとキーが同じサードパーティに保持されないようにするためです。

共有シークレットキーの唯一の要件は、共有シークレットキーを base64 でエンコードして環境変数 C3R_SHARED_SECRET に保存することです。

以下のセクションでは、secret.key ファイルを base64 に変換して環境変数として保存するためのコンソールコマンドについて説明します。secret.key ファイルは、「[ステップ 5: 共有シークレットキーを作成する](#)」で紹介されているコマンドのいずれかによって生成されたもので、ソースの一例にすぎません。

Windows で PowerShell を使用して環境変数にキーを保存

Windows で PowerShell を使用して base64 に変換し、環境変数を設定するには、次のコマンドを実行します。

```
$Bytes=[IO.File]::ReadAllBytes((Get-Location).ToString()+'\secret.key');  
$env:C3R_SHARED_SECRET=[Convert]::ToBase64String($Bytes)
```

Linux または macOS で環境変数にキーを保存

Linux または macOS で base64 に変換し、環境変数を設定するには、次のコマンドを実行します。

```
export C3R_SHARED_SECRET="$(cat secret.key | base64)"
```

ステップ 7: データを暗号化する

このステップを実行するには、AWS Clean Rooms コラボレーション ID と共有シークレットキーを取得する必要があります。詳細については、「[前提条件を満たす](#)」を参照してください。

次の例では、作成した ads.json というスキーマを使用して ads.csv に暗号化を実行します。

データを暗号化するには

1. コラボレーション用の共有シークレットキーを保存します (「[ステップ 6: 共有シークレットキーを環境変数に保存する](#)」を参照)。
2. コマンドラインで、以下のコマンドを入力します。

```
java -jar c3r-cli.jar encrypt <name of input .csv file> --schema=<name of schema .json file> --id=<collaboration id> --output=<name of output.csv file> <optional flags>
```

3. *<name of input .csv file>* には、入力 .csv ファイルの名前を入力します。
4. *schema=* には、.json 暗号化スキーマのファイル名を入力します。
5. *id=* には、コラボレーション ID を入力します。
6. *output=* には、出力ファイルの名前 (例えば `ads-output.csv`) を入力します。
7. 「[暗号コンピューティングパラメータ](#)」および「[Cryptographic Computing for Clean Rooms のオプションフラグ](#)」で説明されているコマンドラインフラグをすべて含めます。
8. コマンドを実行します。

`ads.csv` の例では、以下のコマンドを実行します。

```
java -jar c3r-cli.jar encrypt ads.csv --schema=ads.json --id=123e4567-e89b-42d3-a456-556642440000 --output=ads-output.csv
```

`sales.csv` の例では、以下のコマンドを実行します。

```
java -jar c3r-cli.jar encrypt sales.csv --schema=sales.json --id=123e4567-e89b-42d3-a456-556642440000
```

Note

この例では、出力ファイル名 (`--output=sales-output.csv`) は指定していません。その結果、デフォルトの出力ファイル名 `name-of-file.out.csv` が生成されました。

これで、暗号化されたデータを確認する準備ができました。

ステップ 8: データ暗号化を確認する

データが暗号化されていることを確認するには

1. 暗号化されたデータファイル (sales-output.csv など) を表示します。
2. 以下の列を確認します。

- a. 列 1 – 暗号化済み (username_fingerprint など)。

fingerprint列 (HMAC) には、バージョンとタイプのプレフィックス (01:hmac: など) の後に、base64 でエンコードされた 44 文字のデータがあります。

- b. 列 2 – 暗号化されていません (purchased など)。
- c. 列 3 – 暗号化済み (product_sealed など)。

暗号化された (SELECT) 列では、バージョンとタイプのプレフィックス (01:enc: など) の後に続く、cleartextに任意のパディングを加えた長さは、暗号化されたcleartextの長さに正比例します。つまりこの長さは、入力データのサイズにエンコーディングによる約 33% のオーバーヘッドを加えたものです。

これで次の作業に進むことができます。

1. [暗号化されたデータを S3 にアップロードする](#)
2. [AWS Glue テーブルを作成します。](#)
3. [AWS Clean Roomsで設定済みテーブルを作成する](#)

C3R 暗号化クライアントは、暗号化されていないデータを含まない一時ファイルを作成します (最終出力でそのデータも暗号化されない場合を除く)。ただし、暗号化された値の中には、正しくパディングされていないものもあります。allowRepeatedFingerprintValue のコラボレーション設定が false であっても、フィンガープリント列に重複する値が含まれる場合があります。この問題は、適切なパディング長と重複削除のプロパティがチェックされる前に一時ファイルが書き込まれることが原因で発生します。

暗号化中に C3R 暗号化クライアントに障害や中断が発生すると、一時ファイルを書き込んだ後、これらのプロパティを確認して一時ファイルを削除する前に、C3R 暗号化クライアントが停止することがあります。そのため、こうした一時ファイルがまだディスク上に残っている可能性があります。このような場合、こうしたファイル内のコンテンツでは、プレーンテキストデータが出力と同じレベルで保護されることはありません。特に、このような一時ファイルに対して、最終出力では効果がな

い統計分析が実行されることで、プレーンテキストデータが明らかになる可能性があります。こうしたファイルが権限のない人の手に渡らないように、ユーザーはこれらのファイル (特に SQLite データベース) を削除する必要があります。

(オプション) スキーマの作成 (上級ユーザー)

スキーマの手動作成は、上級ユーザーを対象としています。

以下のセクションでは、列ヘッダーがある場合とない場合の、入力ファイルの JSON スキーマファイル形式について説明しています。上級ユーザーは、必要に応じてスキーマを直接記述または変更できます。

Note

C3R 暗号化クライアントでは、「[例:sealed、fingerprint、およびcleartext列を含む暗号化スキーマの生成](#)」で説明されているインタラクティブなプロセス、またはスタブテンプレートの作成のいずれかを通じてスキーマを作成できます。

マッピングテーブルスキーマと位置テーブルスキーマ

次のセクションでは、2 種類のテーブルスキーマについて説明します。

- マッピングテーブルスキーマ – このスキーマは、ヘッダー行を含む .csv ファイルと Apache Parquet ファイルの暗号化に使用されます。
- 位置テーブルスキーマ – このスキーマは、ヘッダー行のない .csv ファイルの暗号化に使用されます。

C3R 暗号化クライアントでは、コラボレーション用の表形式ファイルを暗号化できます。それには、入力から暗号化された出力をどのように導き出すかを指定するための、対応するスキーマファイルが必要です。

C3R 暗号化クライアントでは、コマンドラインで C3R 暗号化クライアントのスキーマコマンドを実行することで、INPUT ファイルのスキーマを生成できます。java -jar c3r-cli.jar schema --interactive INPUT はコマンドの一例です。

スキーマでは、以下の情報を指定します。

1. ヘッダー名 (マッピングスキーマ) または位置 (位置スキーマ) によって、どのソース列が出力ファイルのどの変換後の列にマッピングされるか
2. どのターゲット列をcleartextのまま残すか
3. SELECT クエリのためにどのターゲット列を暗号化するか
4. JOIN クエリのためにどのターゲット列を暗号化するか

この情報が、テーブル固有の JSON スキーマファイルにエンコードされます。スキーマファイルは headerRow フィールドがブール値の 1 つのオブジェクトで構成されます。この値は、Parquet ファイルとヘッダー行のある .csv ファイルでは true、それ以外の場合は false である必要があります。

マッピングテーブルスキーマ

マッピングスキーマは次のような形式です。

```
{
  "headerRow": true,
  "columns": [
    {
      "sourceHeader": STRING,
      "targetHeader": STRING,
      "type": TYPE,
      "pad": PAD
    },
    ...
  ]
}
```

headerRow が true の場合、オブジェクトの次のフィールドは columns になります。これには、ソースヘッダーをターゲットヘッダーにマッピングする列スキーマの配列 (つまり、出力列に含める内容を記述した JSON オブジェクト) が含まれます。

- sourceHeader – データの取得元となるソース列の STRING ヘッダー名。

Note

同じソース列を複数のターゲット列に使用できます。
スキーマのどこにも sourceHeader としてリストされていない入力ファイルの列は、出力ファイルには表示されません。

- **targetHeader** – 出力ファイル内の対応する列の STRING ヘッダー名。

 Note

このフィールドはマッピングスキーマでは省略可能です。このフィールドを省略すると、sourceHeader が出力ファイルのヘッダー名として再利用されます。出力列が fingerprint 列または sealed 列の場合は、それぞれ `_fingerprint` または `_sealed` が追加されます。

- **type** – 出力ファイルのターゲット列の TYPE。コラボレーションでの列の使用方法に応じて、cleartext、sealed、fingerprint のいずれかになります。
- **pad** – TYPE が sealed の場合にのみ存在する列スキーマオブジェクトのフィールド。対応する値 PAD は、データを暗号化する前にどのようにパディングするかを記述するオブジェクトです。

```
{
  "type": PAD_TYPE,
  "length": INT
}
```

暗号化前のパディングを指定するには、type と length を次のように使用します。

- PAD_TYPE が none の場合 – 列のデータにはパディングが適用されず、length フィールドは無効になります (つまり、省略されます)。
- PAD_TYPE が fixed の場合 – 列のデータは指定された length のバイト数にパディングされます。
- PAD_TYPE が max の場合 – 列のデータは、最も長い値のバイト長に length バイトを加えたサイズにパディングされます。

以下は、各タイプの列を含むマッピングスキーマの例です。

```
{
  "headerRow": true,
  "columns": [
    {
      "sourceHeader": "FullName",
      "targetHeader": "name",
      "type": "cleartext"
    },
    {
```



```

    "sourceHeader": "City",
    "targetHeader": "city_sealed",
    "type": "sealed",
    "pad": {
      "type": "max",
      "length": 16
    }
  },
  {
    "sourceHeader": "PhoneNumber",
    "targetHeader": "phone_number_fingerprint",
    "type": "fingerprint"
  },
  {
    "sourceHeader": "PhoneNumber",
    "targetHeader": "phone_number_sealed",
    "type": "sealed",
    "pad": {
      "type": "fixed",
      "length": 20
    }
  }
]
}

```

より複雑な例として、ヘッダー付きの .csv ファイルの例を以下に示します。

```

FirstName,LastName,Address,City,State,PhoneNumber,Title,Level,Notes
Jorge,Souza,12345 Mills Rd,Anytown,SC,703-555-1234,CEO,10,
Paulo,Santos,0 Street,Anytown,MD,404-555-111,CIO,9,This is a really long note that
could really be a paragraph
Mateo,Jackson,1 Two St,Anytown,NY,304-555-1324,C00,9,""
Terry,Whitlock4 N St,Anytown,VA,407-555-8888,EA,7,Secret notes
Diego,Ramirez,9 Hollows Rd,Anytown,VA,407-555-1222,SDE I,4,null
John,Doe,8 Hollows Rd,Anytown,VA,407-555-4321,SDE I,4,Jane's younger brother
Jane,Doe,8 Hollows Rd,Anytown,VA,407-555-4322,SDE II,5,John's older sister

```

次のマッピングスキーマの例では、列 `FirstName` と `LastName` は cleartext 列です。State 列は fingerprint 列として暗号化され、また、パディングが none で sealed 列としても暗号化されます。残りの列は省略されます。

```

{

```

```

"headerRow": true,
"columns": [
  {
    "sourceHeader": "FirstName",
    "targetHeader": "GivenName",
    "type": "cleartext"
  },
  {
    "sourceHeader": "LastName",
    "targetHeader": "Surname",
    "type": "cleartext"
  },
  {
    "sourceHeader": "State",
    "targetHeader": "State_Join",
    "type": "fingerprint"
  },
  {
    "sourceHeader": "State",
    "targetHeader": "State",
    "type": "sealed",
    "pad": {
      "type": "none"
    }
  }
]
}

```

以下は、マッピングスキーマから生成された .csv ファイルです。

```

givenname,surname,state_fingerprint,state
John,Doe,01:hmac:UK8s8Cn/WR2J0/To2dTxWD73aDEe2ZUXeSHy3Tv
+1Mk=,01:enc:FQ3n3Ahv9BQQNWQGcugeHzHYzEZE1vapHa2Uu4SRgSatZ3q0bjPA4TcsHt
+B0kMKBcnHWI13BeGG/SBqmj7vKpI=
Paulo,Santos,01:hmac:CHF4eIrtTNgAooU9v4h9Qjc
+txBnMidQTjdjWuaDTTA=,01:enc:KZ5n5GtaXACco65AXk48BQ02durDNR2ULc4YxmMC8NaZZKKJiksU1IwFadAvV4iBQ1
Mateo,Jackson,01:hmac:iIRnjfNBzryusIJ1w35lgNzeY1RQ1bSfq6PDHW8Xrbk=,01:enc:mLKpS5HIOSgphdEsrzEd
eN9nB02gAbIygt40Fn4La1Yn9Xyj/XUWX1mn8zFe2T4kyDTD8kG0vpQEUGxAUFk=
Diego,Ramirez,01:hmac:UK8s8Cn/WR2J0/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:rmZhT98Zm
+IIGw1UTjMIJP4IrW/AA1tBLMXcHvnYfRgmWP623VFQ6aUnhsb2MDqEw4G5Uwg5rKKZepUxx5uKbfk=
Jorge,Souza,01:hmac:3BxJdXiFFyZ8HBbYNqqEhBVqhN0d7s2ZiKUe7QiTy08=,01:enc:vVaqWC1VRbhvkf8gnuR7q0z
Terry,Whitlock01:hmac:UK8s8Cn/WR2J0/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:3c9VEWb0D0/
xbQjdGuccLvI7oZTBdPU+SyrJIyr2kudfAxbuMQ2uRdU/q7rbgyJjxZS8M2U35ILJf/1DgTyg7cM=

```

```
Jane,Doe,01:hmac:UK8s8Cn/WR2J0/To2dTxD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:9RWv46YLveykeNZ/G0Nd1YFg+AVd0nu05hHyAYTQkPLHnyX+0/jbzD/g9ZT8GCgVE9aB5bV4ooJIXHGBVMXcjrQ=
```

位置テーブルスキーマ

位置スキーマは次のような形式です。

```
{
  "headerRow": false,
  "columns": [
    [
      {
        "targetHeader": STRING,
        "type": TYPE,
        "pad": PAD
      },
      {
        "targetHeader": STRING,
        "type": TYPE,
        "pad": PAD
      }
    ],
    [],
    ...
  ]
}
```

headerRow が false の場合、オブジェクトの次のフィールドはエントリの配列を含む columns になります。各エントリは、それ自体が 0 個以上の位置列スキーマ (sourceHeader フィールドなし) の配列、つまり、出力に含める内容を記述した JSON オブジェクトです。

- sourceHeader – データの取得元となるソース列の STRING ヘッダー名。

Note

位置スキーマでは、このフィールドは省略する必要があります。位置スキーマの場合、ソース列は、スキーマファイルの列の対応するインデックスによって推測されます。

- targetHeader – 出力ファイル内の対応する列の STRING ヘッダー名。

Note

このフィールドは、位置スキーマでは必須です。

- **type** – 出力ファイルのターゲット列の TYPE。コラボレーションでの列の使用方法に応じて、cleartext、sealed、fingerprint のいずれかになります。
- **pad** – TYPE が sealed の場合にのみ存在する列スキーマオブジェクトのフィールド。対応する値 PAD は、データを暗号化する前にどのようにパディングするかを記述するオブジェクトです。

```
{
  "type": PAD_TYPE,
  "length": INT
}
```

暗号化前のパディングを指定するには、type と length を次のように使用します。

- PAD_TYPE が none の場合 – 列のデータにはパディングが適用されず、length フィールドは無効になります (つまり、省略されます)。
- PAD_TYPE が fixed の場合 – 列のデータは指定された length のバイト数にパディングされます。
- PAD_TYPE が max の場合 – 列のデータは、最も長い値のバイト長に length バイトを加えたサイズにパディングされます。

Note

fixed は、列のデータのバイトサイズの上限が事前にわかっている場合に便利です。その列に指定した length 値よりも長いデータが 1 つでもあると、エラーが発生します。max は、データのサイズに関係なく動作するため、入力データの正確なサイズが不明な場合に便利です。ただし max の場合は、データが 2 回暗号化されるため、処理時間が長くなります。max では、データを一時ファイルに読み込むときに 1 回暗号化し、列内の最も長いデータエントリがわかった後に 1 回暗号化します。

また、最も長い値の長さは、クライアントの呼び出しの間には保存されません。データをバッチで暗号化したり、新しいデータを定期的に暗号化したりする場合は、生成される暗号文の長さはバッチによって異なる可能性があることに注意してください。

以下は位置スキーマの例です。

```
{
  "headerRow": false,
  "columns": [
    [
      {
        "targetHeader": "name",
        "type": "cleartext"
      }
    ],
    [
      {
        "targetHeader": "city_sealed",
        "type": "sealed",
        "pad": {
          "type": "max",
          "length": 16
        }
      }
    ],
    [
      {
        "targetHeader": "phone_number_fingerprint",
        "type": "fingerprint"
      },
      {
        "targetHeader": "phone_number_sealed",
        "type": "sealed",
        "pad": {
          "type": "fixed",
          "length": 20
        }
      }
    ]
  ]
}
```

複雑な例として、先頭にヘッダー行がない .csv ファイルの例を以下に示します。

```
Jorge,Souza,12345 Mills Rd,Anytown,SC, 703 -555 -1234,CEO, 10,
Paulo,Santos, 0 Street,Anytown,MD, 404-555-111,CIO, 9,This is a really long note that
could really be a paragraph
Mateo,Jackson, 1 Two St,Anytown,NY, 304-555-1324,C00, 9, ""
Terry,Whitlock, 4 N St,Anytown,VA, 407-555-8888,EA, 7,Secret notes
```

```
Diego,Ramirez, 9 Hollows Rd,Anytown,VA, 407-555-1222,SDE I, 4,null
John,Doe, 8 Hollows Rd,Anytown,VA, 407-555-4321,SDE I, 4,Jane's younger brother
Jane,Doe, 8 Hollows Rd,Anytown,VA, 407-555-4322,SDE II, 5,John's older sister
```

位置スキーマは次のような形式になります。

```
{
  "headerRow": false,
  "columns": [
    [
      {
        "targetHeader": "GivenName",
        "type": "cleartext"
      }
    ],
    [
      {
        "targetHeader": "Surname",
        "type": "cleartext"
      }
    ],
    [],
    [],
    [
      {
        "targetHeader": "State_Join",
        "type": "fingerprint"
      },
      {
        "targetHeader": "State",
        "type": "sealed",
        "pad": {
          "type": "none"
        }
      }
    ],
    [],
    [],
    [],
    []
  ]
}
```

前述のスキーマは、指定されたターゲットヘッダーがヘッダー行に含まれる出力ファイルを生成します。

```
givenname,surname,state_fingerprint,state
Mateo,Jackson,01:hmac:iIRnjfNBzryusIJ1w35lgNzeY1RQ1bSfq6PDHW8Xrbk=,01:enc:ENS6QD3cMV19vQEGfe9MM
Q8m/Y5SA89dJwKpT5rGPp8e36h6klwDoslpFzGvU0=
Jorge,Souza,01:hmac:3BxJdXiFFyZ8HBbYNqqEhBVqhN0d7s2ZiKUe7QiTy08=,01:enc:LKo0zirq2+
+XEIIIMNRjAsGMdyWUDwYaum0B+IFP+rUf1BNeZDJjtFe1Z+zbZfXQWwJy52Rt7HqvAb2WIK1oMmk=
Paulo,Santos,01:hmac:CHF4eIrtTNgAooU9v4h9Qjc
+txBnMidQTjdjWuaDTTA=,01:enc:MyQKyWxJ9kvK1xDQQtX1UNwv3F+yRBRr0xrUY/1BGg5KFg0n9pK+MZ7g
+ZNqZEPcPz4lht1u0t/wbTaqz0CLXFQ=
Jane,Doe,01:hmac:UK8s8Cn/WR2J0/To2dTxD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:Pd8sbITBfb0/
ttUB4svVsGoYkDfnDvgkvxzeCi0Yxq54rLSwccy1o3/B50C3cpkkn56dovCwzgmmpNwrmCmYtb4=
Terry,Whitlock01:hmac:UK8s8Cn/WR2J0/To2dTxD73aDEe2ZUXeSHy3Tv
+1Mk=,01:enc:Qmtzu3B3GAXKh2KkRYTiEAaMopYedsSdF2e/
ADUiBQ9kv2CxKpZWyYTD3ztmKPMka19dHre5VhUHNp030+j1AQ8=
Diego,Ramirez,01:hmac:UK8s8Cn/WR2J0/To2dTxD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:ysdg
+GHKdeZrS/geBIoo0EPLHG68MsWpx1dh3xjb+fG5rmFmqUcJLNuuYBHhHA1xchM2WVeV1fmHkBX3mvZNvkc=
John,Doe,01:hmac:UK8s8Cn/WR2J0/To2dTxD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:9uX0wZu07kAPAx
+Hf6uvQownkWqFSKtWS7gQIJSe5aXFquKWCK6yZN0X5Ea2N3bn03Uj1kh0agDWoiP9FRZGJA4=
```

C3R 暗号化クライアントによるデータテーブルの復号化

Cryptographic Computing for Clean Rooms と C3R 暗号化クライアントを使用してデータテーブルを暗号化するコラボレーションでは、以下の手順に従ってください。この手順は、[コラボレーションでデータにクエリを実行](#)した後に使用します。

この手順には、共有シークレットキーとコラボレーション ID が必要です。

結果を受け取れるメンバーは、コラボレーションのデータを暗号化するときに使用されたものと同じ共有シークレットキーとコラボレーション ID を使用してデータを復号化します。

Note

AWS Clean Rooms コラボレーションでは、クエリ結果を実行および表示できるユーザーは既に制限されています。結果にアクセスできるユーザーが誰であっても、復号化を実行するには、データの暗号化に使用されたものと同じ共有シークレットキーとコラボレーション ID が必要になります。

暗号化されたデータテーブルを復号化するには

1. (オプション) [C3R 暗号化クライアントで使用可能なコマンドを確認](#)します。
2. (オプション) 目的のディレクトリに移動し、ls (macOS) または dir (Windows) を実行します。
 - c3r-cli.jar ファイルと暗号化されたクエリ結果データファイルが目的のディレクトリにあることを確認します。

Note

クエリ結果が AWS Clean Rooms コンソールインターフェイスからダウンロードされた場合、ユーザーアカウントのダウンロードフォルダにある可能性があります。(例えば、Windows および macOS のユーザーディレクトリにある [ダウンロード] フォルダ)。クエリ結果ファイルは c3r-cli.jar と同じフォルダに移動することをお勧めします。

3. 共有シークレットキーを C3R_SHARED_SECRET 環境変数に保存します。詳細については、「[ステップ 6: 共有シークレットキーを環境変数に保存する](#)」を参照してください。
4. AWS Command Line Interface (AWS CLI) から、次のコマンドを実行します。

```
java -jar c3r-cli.jar decrypt <name of input .csv file> --id=<collaboration id> --  
output=<output file name>
```

5. 各#####を独自の情報に置き換えます。
 - a. id= には、コラボレーション ID を入力します。
 - b. output= には、出力ファイルの名前 (例えば results-decrypted.csv) を入力します。

出力名を指定しないと、デフォルトの名前がターミナルに表示されます。

- c. CSV または Parquet 表示用の任意のアプリケーション (Microsoft Excel、テキストエディタ、その他のアプリケーションなど) を使用して、指定した出力ファイル内の復号化されたデータを表示します。

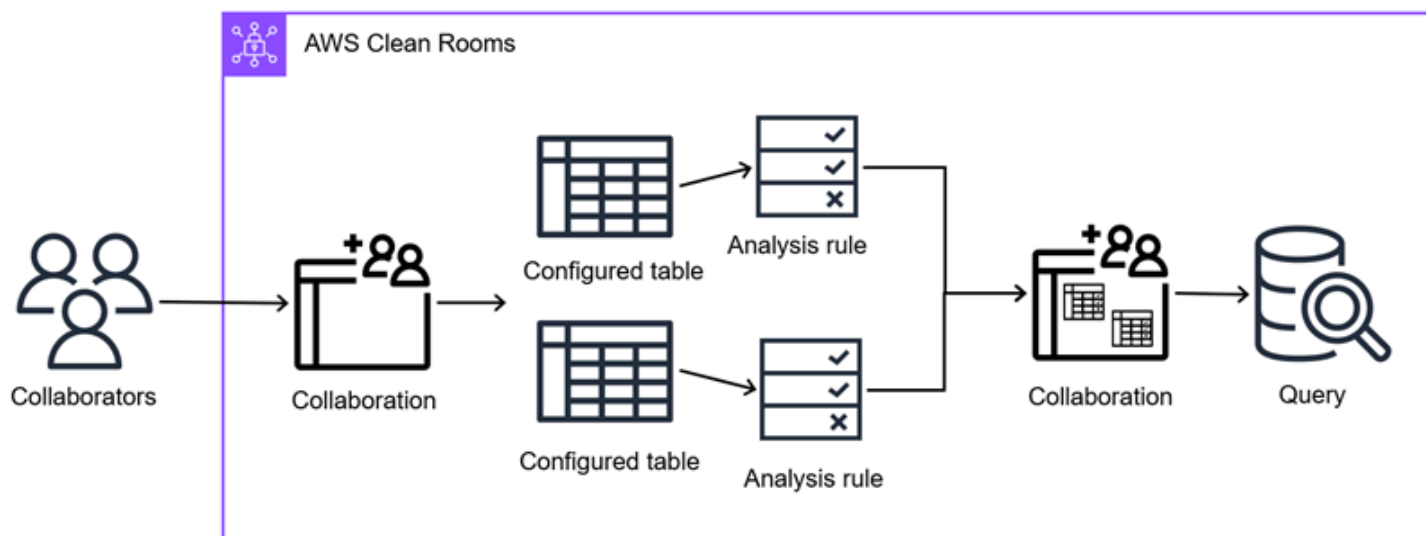
の設定済みテーブル AWS Clean Rooms

設定済みテーブルは、データソース内の既存のテーブルへの参照です。これには、データのクエリ AWS Clean Rooms 方法を決定する分析ルールが含まれており、テーブルの使用状況を制御するためのデータアクセス予算を含めることができます。設定済みテーブルは 1 つ以上のコラボレーションに関連付けることができます。

を使用すると AWS Clean Rooms、購入数と購入数の比較など、イベントデータの集約分析を実行できます。また、重複する顧客データをセグメントデータから CRM データに強化するなど、イベントデータのリスト分析を実行することもできます。さらに、カスタムクエリを実行し、表示者データやセグメント属性などのイベントデータに差分プライバシーを設定することもできます。これらの分析タイプのいずれかについて、データアクセス予算を設定して、クエリを介してアクセスされるデータの量を監視および制御できます。

まず、でコラボレーションを作成し AWS Clean Rooms、招待 AWS アカウント する を追加するか、メンバーシップを作成して招待されたコラボレーションに参加します。次に、コラボレーションの他のメンバーと共に設定済みテーブルを作成します。設定済みテーブル (集計、リスト、カスタム) に分析ルールを追加し、オプションでデータアクセス予算を設定します。次に、設定されたテーブルをコラボレーションに関連付けます。最後に、クエリを実行できるメンバーは 2 つのデータテーブルでクエリを実行し、クエリの実行時にデータアクセス予算を消費します。

次の図は、でイベントデータを操作する方法をまとめたものです AWS Clean Rooms。



トピック

- [で設定済みテーブルを作成する AWS Clean Rooms](#)

- [設定済みテーブルへの分析ルールの追加](#)
- [設定済みテーブルのコラボレーションへの関連付け](#)
- [Configuring a data access budget](#)
- [設定済みテーブルへのコラボレーション分析ルールの追加](#)
- [差分プライバシーポリシーの設定 \(オプション\)](#)
- [テーブルと分析ルールの表示](#)
- [設定済みテーブルの編集](#)
- [設定済みテーブルのタグの編集](#)
- [設定済みテーブル分析ルールの編集](#)
- [設定済みテーブル分析ルールの削除](#)
- [設定済みテーブルの許可されていない列](#)
- [設定済みテーブルの関連付けの編集](#)
- [設定済みテーブルの関連付けの解除](#)

で設定済みテーブルを作成する AWS Clean Rooms

設定済みテーブルは、データソース内の既存のテーブルへの参照です。設定済みテーブルには、AWS Clean Roomsでどのようにデータにクエリを実行するかを決定する分析ルールが含まれています。設定済みテーブルは 1 つ以上のコラボレーションに関連付けることができます。

AWS SDKs を使用して設定済みテーブルを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

トピック

- [設定済みテーブルの作成 – Amazon S3 データソース](#)
- [設定済みテーブルの作成 – Amazon Athena データソース](#)
- [設定済みテーブルの作成 – Snowflake データソース](#)

設定済みテーブルの作成 – Amazon S3 データソース

この手順では、[メンバー](#)は次のタスクを実行します。

- で使用する既存の AWS Glue テーブルを設定します AWS Clean Rooms。(このステップは、Cryptographic Computing for Clean Rooms を使用している場合を除き、コラボレーションに参加する前でも後でも実行できます)。

Note

AWS Clean Rooms は AWS Glue テーブルをサポートします。データの取得の詳細については AWS Glue、「」を参照してください [ステップ 3: データテーブルを Amazon S3 にアップロードする](#)。

- [設定済みテーブル](#)に名前を付け、コラボレーションに使用する列を選択します。

次の手順は、次のことを前提としています。

- コラボレーションメンバーは、データ [テーブルを Amazon S3 にアップロードし](#)、[AWS Glue テーブルを作成済み](#)です。

Note

Amazon S3 の結果の送信先をデータソースと同じ S3 バケット内にすることはできません。

- (オプション) [暗号化](#)されたデータテーブルの場合は、コラボレーションメンバーが、C3R 暗号化クライアントを使用して既に [暗号化されたデータテーブルを準備](#)していること。

によって提供される統計生成を使用して AWS Glue、AWS Glue Data Catalog テーブルの列レベルの統計を計算できます。がデータカタログ内のテーブルの統計 AWS Glue を生成すると、Amazon Redshift Spectrum はそれらの統計を自動的に使用してクエリプランを最適化します。を使用した列レベルの統計の計算の詳細については AWS Glue、AWS Glue 「ユーザーガイド」の「[列統計を使用したクエリパフォーマンスの最適化](#)」を参照してください。詳細については AWS Glue、[AWS Glue デベロッパーガイド](#)を参照してください。

設定済みテーブルを作成するには – Amazon S3 データソース

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。

3. 右上隅にある [新しいテーブルを設定] を選択します。
4. データソースで、AWS データソースで Amazon S3 を選択します。
5. Amazon S3 テーブルの下:
 - a. S3 テーブルがホストされているリージョンを選択します。

デフォルトでは、現在のリージョン (バージニア北部 us-east-1 など) が選択されます。

 Warning

Amazon S3 データソースが処理場所とは異なるリージョンにある場合、データ処理はソースリージョンの外部で一時的に発生する可能性があります。先に進む前に、クロスリージョンのデータ移動がデータ主権要件、規制コンプライアンスポリシー、データガバナンス標準に準拠していることを確認します。

リージョンの詳細については、の「[リージョンとエンドポイント](#)」を参照してくださいAWS 全般のリファレンス。

- b. ドロップダウンリストからデータベースを選択します。
- c. 設定する [テーブル] をドロップダウンリストから選択します。

 Note

テーブルが正しいことを確認するには、次のいずれかの操作を行います。

- 「表示 AWS Glue」を選択します。
- スキーマを表示するには、 から AWS Glueスキーマの表示を有効にします。

 Important

データが CSV 形式の AWS Glue テーブルの場合、Glue スキーマの列名と順序は CSV データと完全に一致する必要があります。一致しない場合、設定済みテーブルの許可された列リストが正しく適用されない可能性があります。

6. コラボレーションで許可される列と分析方法の場合、

a. コラボレーションで許可する列

- すべての列を選択して、コラボレーションですべての列をクエリできるようにします。
- カスタムリストを選択して、許可された列を指定ドロップダウンリストから 1 つ以上の列をコラボレーションでクエリできるようにします。

b. 許可された分析方法の場合、

- i. Direct query を選択して、このテーブルで SQL クエリを直接実行できるようにします。
- ii. Direct job を選択して、このテーブルで PySpark ジョブを直接実行できるようにします。

Example例

たとえば、コラボレーションメンバーがすべての列で直接 SQL クエリと PySpark ジョブの両方を実行できるようにする場合は、すべての列、直接クエリ、および直接ジョブを選択します。

7. [設定済みのテーブルの詳細] で以下の操作を行います。

a. 設定済みテーブルの [名前] を入力します。

デフォルトの名前を使用することも、テーブルの名前を変更することもできます。

b. テーブルの [説明] を入力します。

この説明は、似たような名前を持つ他の設定済みテーブルと区別するのに役立ちます。

8. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。

9. [新しいテーブルを設定] を選択します。

設定済みテーブルを作成したら、次の作業に進むことができます。

- [設定済みテーブルへの分析ルールの追加](#)
- [設定済みテーブルをコラボレーションに関連付ける](#)

設定済みテーブルの作成 – Amazon Athena データソース

Amazon Athena データソースオプションを使用すると、Amazon S3 に保存され、データカタログまたはフェデレーティッドカタログにカタログ化された AWS Glue データをクエリし、を介してアクセスを制御できます AWS Lake Formation。テーブルと AWS Glue Data Catalog ビューの両方がサポートされています。Lake Formation リソースリンクを使用して、テーブルとビュー AWS リージョン を AWS Clean Rooms コラボレーションに結合する AWS Clean Rooms メンバーアカウント AWS アカウント 間で共有できます。

Note

Athena データソース統合を介してクエリできるのは、Amazon S3 ベースのデータセットのみです。

この手順では、[メンバー](#)は次のタスクを実行します。

- を使用する AWS Glue Data Catalog ように の既存のテーブルまたはビューを設定します。 AWS Clean Rooms
- [設定済みテーブル](#)に名前を付け、コラボレーションに使用する列を選択します。

次の手順は、次のことを前提としています。

- コラボレーションメンバーは AWS Glue Data Catalog 、データベースとテーブルまたは GDC ビューを既に作成しています。

設定済みテーブルを作成するには – Athena データソース

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 右上隅にある [新しいテーブルを設定] を選択します。
4. データソースで、AWS データソースで Amazon Athena を選択します。
5. Amazon Athena テーブルの下:
 - a. Amazon Athena テーブルがホストされているリージョンを選択します。

デフォルトでは、現在のリージョン (バージニア北部 us-east-1 など) が選択されます。

 Warning

Amazon Athena データソースが処理場所とは異なるリージョンにある場合、データ処理はソースリージョンの外部で一時的に発生する可能性があります。先に進む前に、クロスリージョンのデータ移動がデータ主権要件、規制コンプライアンスポリシー、データガバナンス標準に準拠していることを確認します。

リージョンの詳細については、の「[リージョンとエンドポイント](#)」を参照してくださいAWS 全般のリファレンス。

- b. ドロップダウンリストからカタログを選択します。

デフォルトでは、AWS Glue データカタログが選択されます。

- AWS Glue データカタログ – のテーブルのデフォルトカタログ AWS Glue。
- フェデレーティッドカタログ – リモート Apache Iceberg REST カタログに接続するように AWS Glue カタログフェデレーションを設定している場合に使用できます。詳細については、「AWS Lake Formation デベロッパーガイド」の「[カタログフェデレーション](#)」を参照してください。

- c. ドロップダウンリストからデータベースを選択します。
- d. 設定する [テーブル] をドロップダウンリストから選択します。

 Note

テーブルが正しいことを確認するには、次のいずれかの操作を行います。

- で表示 AWS Glue または で表示 AWS Lake Formation (カタログタイプに応じて) を選択します。
- スキーマを表示するには、 から AWS Glueスキーマの表示を有効にします。

6. Amazon Athena 設定の場合、

- a. ドロップダウンリストからワークグループを選択します。

- b. S3 出力の場所では、次のいずれかのシナリオに基づいて、推奨されるアクションを選択します。

シナリオ	推奨されるアクション
ワークグループにはデフォルトの出力場所がありません。	S3 出力場所を入力するか、S3 を参照を選択します。
ワークグループはデフォルトの出力場所を適用します。	S3 出力場所は自動的に選択され、変更することはできません。
ワークグループはデフォルトの出力場所を適用しません。	S3 出力場所を入力するか、Browse S3 を選択します。

7. コラボレーションで許可される列については、目標に基づいてオプションを選択します。

目標	推奨されるオプション
すべての列の使用を許可する AWS Clean Rooms (分析ルールに従う)	すべての列
許可された列を指定ドロップダウンリストから 1 つ以上の列を許可する	カスタムリスト

8. [設定済みのテーブルの詳細] で以下の操作を行います。

- a. 設定済みテーブルの [名前] を入力します。

デフォルトの名前を使用することも、テーブルの名前を変更することもできます。

- b. テーブルの [説明] を入力します。

この説明は、似たような名前を持つ他の設定済みテーブルと区別するのに役立ちます。

- c. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。

9. [新しいテーブルを設定] を選択します。

設定済みテーブルを作成したら、次の作業に進むことができます。

- [設定済みテーブルへの分析ルールの追加](#)
- [設定済みテーブルをコラボレーションに関連付ける](#)

設定済みテーブルの作成 – Snowflake データソース

この手順では、[メンバー](#)は次のタスクを実行します。

- で使用する既存の Snowflake テーブルを設定します AWS Clean Rooms。(このステップは、Cryptographic Computing for Clean Rooms を使用している場合を除き、コラボレーションに参加する前でも後でも実行できます)。
- [設定済みテーブル](#)に名前を付け、コラボレーションに使用する列を選択します。

次の手順は、次のことを前提としています。

- コラボレーションメンバーはすでにデータテーブルを Snowflake にアップロードしています。
- (オプション) [暗号化](#)されたデータテーブルの場合は、コラボレーションメンバーが、C3R 暗号化クライアントを使用して既に[暗号化されたデータテーブルを準備](#)していること。

設定済みテーブルを作成するには – Snowflake データソース

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 右上隅にある [新しいテーブルを設定] を選択します。
4. データソースの場合、サードパーティーのクラウドとデータソースで Snowflake を選択します。
5. 既存のシークレット ARN を使用するか、このテーブルの新しいシークレットを保存して、Snowflake 認証情報を指定します。

Use existing secret ARN

1. シークレット ARN がある場合は、シークレット ARN フィールドに入力します。

Go to AWS Secrets Manager を選択すると、シークレット ARN を検索できます。

2. 別のテーブルから既存のシークレットがある場合は、既存のテーブルからシークレット ARN をインポートを選択します。

 Note

シークレット ARN はクロスアカウントとすることができます。

Store a new secret for this table

1. 次の Snowflake 認証情報を入力します。
 - Snowflake ユーザー名
 - Snowflake ウェアハウス
 - Snowflake ロール
 - Snowflake Privacy Enhanced Mail (PEM) プライベートキー
 2. 暗号化の場合は、次のいずれかを実行します。
 - AWS マネージドキー (デフォルト) を使用するには、暗号化設定のカスタマイズチェックボックスをオフにしたままにします。
 - カスタムを使用するには AWS KMS key:
 - 暗号化設定をカスタマイズチェックボックスをオンにします。
 - KMS キーの場合は、キー ARN を入力するか、リストから選択します。
 3. 後で認証情報を見つけるのに役立つシークレット名を入力します。
6. Snowflake テーブルとスキーマの詳細には、詳細を手動で入力するか、詳細を自動的にインポートします。

Enter the details manually

1. Snowflake アカウント識別子を入力します。

詳細については、Snowflake ドキュメントの「[アカウント識別子](#)」を参照してください。

アカウント識別子は、Snowflake ドライバーに使用される形式である必要があります。識別子が の形式になるように、ピリオド (.) をハイフン (-) に置き換える必要があります <orgname>-<account_name>。

2. Snowflake データベースを入力します。

詳細については、[Snowflake ドキュメントの「Snowflake データベース」](#)を参照してください。

3. Snowflake スキーマ名を入力します。
4. Snowflake テーブル名を入力します。

詳細については、[Snowflake ドキュメントの「Snowflake テーブル構造について」](#)を参照してください。

5. スキーマには、列名を入力し、ドロップダウンリストからデータ型を選択します。
6. 列の追加を選択して列を追加します。
 - オブジェクトデータ型を選択した場合は、オブジェクトスキーマを指定します。

Exampleオブジェクトスキーマの例

```
name STRING,  
location OBJECT(  
  x INT,  
  y INT,  
  metadata OBJECT(uuid STRING)  
),  
history ARRAY(TEXT)
```

- 配列データ型を選択した場合は、配列スキーマを指定します。

Example配列スキーマの例

```
OBJECT(x INT, y INT)
```

- Map データ型を選択した場合は、Map スキーマを指定します。

Exampleマップスキーマの例

```
STRING, OBJECT(x INT, y INT)
```

Automatically import the details

1. Snowflake から COLUMNS ビューを CSV ファイルとしてエクスポートします。

Snowflake COLUMNS ビューの詳細については、Snowflake ドキュメントの「[COLUMNS ビュー](#)」を参照してください。

2. ファイルからインポートを選択して CSV ファイルをインポートし、追加情報を指定しま

す。

データベース名、スキーマ名、テーブル名、列名、データ型が自動的にインポートされます。

- オブジェクトデータ型を選択した場合は、オブジェクトスキーマを指定します。
- 配列データ型を選択した場合は、配列スキーマを指定します。
- Map データ型を選択した場合は、Map スキーマを指定します。

3. Snowflake アカウント識別子を入力します。

詳細については、Snowflake ドキュメントの「[アカウント識別子](#)」を参照してください。

 Note

テーブルスキーマを自動的に取得するには AWS Glue 、でカタログ化された S3 テーブルのみを使用できます。

7. コラボレーションで許可される列については、目標に基づいてオプションを選択します。

目標	推奨されるオプション
すべての列の使用を許可する AWS Clean Rooms (分析ルールに従う)	すべての列
許可された列を指定ドロップダウンリストから 1 つ以上の列を許可する	カスタムリスト

8. [設定済みのテーブルの詳細] で以下の操作を行います。

a. 設定済みテーブルの [名前] を入力します。

デフォルトの名前を使用することも、テーブルの名前を変更することもできます。

b. テーブルの [説明] を入力します。

この説明は、似たような名前を持つ他の設定済みテーブルと区別するのに役立ちます。

c. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。

9. [新しいテーブルを設定] を選択します。

設定済みテーブルを作成したら、次の作業に進むことができます。

- [設定済みテーブルへの分析ルールの追加](#)
- [設定済みテーブルをコラボレーションに関連付ける](#)

設定済みテーブルへの分析ルールの追加

以下のセクションでは、設定済みテーブルに分析ルールを追加する方法について説明します。分析ルールを定義すると、クエリを行えるメンバーに対して、AWS Clean Roomsでサポートされる特定の分析ルールに一致するクエリを実行する権限を与えることができます。

AWS Clean Rooms は、次のタイプの分析ルールをサポートしています。

- [集計分析ルール](#)
- [リスト分析ルール](#)
- [のカスタム分析ルール AWS Clean Rooms](#)

設定済みテーブルごとに 1 つの分析ルールのみを割り当てることができます。設定済みテーブルをコラボレーションに関連付ける前に、いつでも分析ルールを設定できます。

Important

Cryptographic Computing for Clean Rooms を使用しており、コラボレーション内のデータテーブルを暗号化している場合、暗号化された設定済みテーブルに追加する分析ルールは、データの暗号化方法と一致している必要があります。例えば、SELECT のデータ (集計分析ルール) を暗号化した場合、JOIN の分析ルール (リスト分析ルール) は追加しないでください。

トピック

- [集計分析ルールのテーブルへの追加 \(ガイドフロー\)](#)
- [リスト分析ルールのテーブルへの追加 \(ガイドフロー\)](#)
- [カスタム分析ルールのテーブルへの追加 \(ガイドフロー\)](#)
- [テーブルへの分析ルールの追加 \(JSON エディタ\)](#)
- [次の手順](#)

集計分析ルールのテーブルへの追加 (ガイドフロー)

集計分析ルールでは、COUNT、SUM、および AVG 関数を使用して、行レベルの情報を明らかにすることなく、任意のディメンションで統計を集計するクエリが可能になります。

この手順では、AWS Clean Rooms コンソールの [ガイドフロー] オプションを使用して設定済みテーブルに集計分析ルールを追加するプロセスを説明します。

Note

non-S3データソースを使用する設定済みテーブルは、[カスタム分析ルール](#)のみをサポートします。

集計分析ルールをテーブルに追加するには (ガイドフロー)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[分析ルールを設定] を選択します。
5. ステップ 1: 分析ルールタイプを選択し、分析ルールタイプで集計オプションを選択します。
6. [作成方法] で [ガイドフロー] を選択し、[次へ] を選択します。
7. [ステップ 2: クエリコントロールを指定] の [集約関数] で以下の操作を行います。
 - a. ドロップダウンから集約関数を選択します。
 - COUNT
 - COUNT DISTINCT
 - SUM
 - SUM DISTINCT
 - AVG
 - b. 集約関数で使用可能にする列を [列] ドロップダウンから選択します。
 - c. (オプション) [別の関数を追加] を選択して別の集約関数を追加し、その関数に 1 つ以上の列を関連付けます。

 **Note**

少なくとも 1 つの集約関数が必要です。

- d. (オプション) [削除] を選択して集約関数を削除します。
8. [結合コントロール] で以下の操作を行います。
- a. [テーブルの単独クエリを許可] オプションを 1 つ選択します。

選択内容	結果
いいえ、クエリできるのは重複部分だけです	テーブルにクエリを実行できるのは、クエリを行えるメンバーが所有するテーブルに結合されている場合のみになります。
あり	テーブルは単独でクエリを実行することも、他のテーブルに結合している場合にクエリを実行することもできます。

- b. [結合列を指定] で、INNER JOIN ステートメントでの使用を許可する列を選択します。
- 前のステップで [はい] を選択した場合、これは省略可能です。
- c. [照合に使用できる演算子を指定してください] で、複数の結合列の照合で使用可能にする演算子 (ある場合) を選択します。2 つ以上の JOIN 列を選択した場合、これらの演算子のいずれかが必要です。

選択内容	結果
AND	INNER JOIN 一致条件に AND を含めて、テーブル間で 1 つの列を別の列に結合できます。
または	INNER JOIN 一致条件に OR を含めて、テーブル間での複数の列一致を組み合わせることができます。この論理演算子は一致率を高めるのに便利です。

9. (オプション) [ディメンションコントロール] の [ディメンション列を指定] ドロップダウンで、SELECT ステートメントと、クエリの WHERE、GROUP BY、および ORDER BY 部分での使用を許可する列を選択します。

 Note

集約関数または結合列はディメンション列として使用できません。

10. [スカラー関数] の [許可するスカラー関数] でオプションを 1 つ選択します。

選択内容	結果
で現在サポートされているすべての AWS Clean Rooms	AWS Clean Roomsで現在サポートされているすべてのスカラー関数を許可します。 • [リストを表示] を選択すると、[AWS Clean Roomsでサポートされているスカラー関数] の全リストが表示されます。
カスタムリスト	どのスカラー関数を許可するかをカスタマイズできます。 • [許可されるスカラー関数を指定] ドロップダウンから 1 つまたは複数のオプションを選択します。
[なし]	スカラー関数を許可しません。

詳細については、「[スカラー関数](#)」を参照してください。

11. [次へ] を選択します。
12. [ステップ 3: クエリ結果コントロールを指定] の [集約制約] で以下の操作を行います。
- 各 [列名] のドロップダウンリストを選択します。
 - COUNT DISTINCT 関数を適用した後に、返される各出力行が満たす必要がある [個別値の最小数] をドロップダウンリストから選択します。
 - [制約を追加] を選択して集約制約を追加します。
 - (オプション) [削除] を選択して集約制約を削除します。

13. [追加分析が出力に適用されました] で、目標に応じてオプションを選択します。

目標	推奨されるオプション
このテーブルで直接クエリのみを許可します。クエリ結果に対する追加分析の実行を拒否します。テーブルは直接クエリにのみ使用できます。	許可されていません
このテーブルで直接クエリと追加分析の両方を許可しますが、必須にはしません。	許可されています
テーブルは、必要な追加分析のいずれかで処理される直接クエリでのみ使用できることを要求します。このテーブルの直接クエリは、返す前にさらに処理する必要があります。	必須

14. [次へ] を選択します。

15. [ステップ 4: 確認して設定] で、前のステップで行った選択内容を確認し、必要に応じて編集して、[分析ルールを設定] を選択します。

集計分析ルールがテーブルに正常に設定されたことを示す確認メッセージが表示されます。

リスト分析ルールのテーブルへの追加 (ガイドフロー)

リスト分析ルールでは、クエリを実行できるメンバーのテーブルと関連するテーブルとの重複部分について行レベルのリストを出力するクエリが可能になります。

この手順では、AWS Clean Rooms コンソールのガイド付きフローオプションを使用して、リスト分析ルールを設定済みテーブルに追加するプロセスについて説明します。

Note

non-S3データソースを使用する設定済みテーブルは、[カスタム分析ルール](#)のみをサポートします。

リスト分析ルールをテーブルに追加するには (ガイドフロー)

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[分析ルールを設定] を選択します。
5. ステップ 1: 分析ルールタイプを選択し、分析ルールタイプでリストオプションを選択します。
6. [作成方法] で [ガイドフロー] を選択し、[次へ] を選択します。
7. [ステップ 2: クエリコントロールを指定] の [結合コントロール] で以下の操作を行います。
 - a. [結合列を指定] で、INNER JOIN ステートメントでの使用を許可する列を選択します。
 - b. [照合に使用できる演算子を指定してください] で、複数の結合列の照合で使用可能にする演算子 (ある場合) を選択します。2 つ以上の JOIN 列を選択した場合、これらの演算子のいずれかが必要です。

選択内容	結果
AND	INNER JOIN 一致条件に AND を含めて、テーブル間で 1 つの列を別の列に結合できます。
または	INNER JOIN 一致条件に OR を含めて、テーブル間での複数の列一致を組み合わせることができます。この論理演算子は一致率を高めるのに便利です。

8. (オプション) [リストコントロール] の [リスト列を指定] ドロップダウンで、クエリ出力での使用 (つまり、SELECT ステートメントでの使用)、または結果のフィルター処理での使用 (つまり WHERE ステートメントでの使用) を許可する列を選択します。
9. [次へ] を選択します。
10. [ステップ 3: クエリ結果コントロールを指定] の [追加分析が出力に適用されました] で、目標に応じてオプションを選択します。

目標	推奨されるオプション
このテーブルで直接クエリのみを許可します。クエリ結果に対する追加分析の実行を拒否します。テーブルは直接クエリにのみ使用できます。	許可されていません
このテーブルで直接クエリと追加分析の両方を許可しますが、必須にはしません。	許可されています
テーブルは、必要な追加分析のいずれかで処理される直接クエリでのみ使用できることを要求します。このテーブルの直接クエリは、返す前にさらに処理する必要があります。	必須

11. [ステップ 4: 確認して設定] で、前のステップで行った選択内容を確認し、必要に応じて編集して、[分析ルールを設定] を選択します。

リスト分析ルールがテーブルに正常に設定されたことを示す確認メッセージが表示されます。

カスタム分析ルールのテーブルへの追加 (ガイドフロー)

カスタム分析ルールは、設定されたテーブルでカスタム SQL クエリまたは PySpark ジョブを有効にします。カスタム分析ルールは、以下を使用している場合に必要になります。

- 事前に承認された特定の SQL クエリまたは PySpark ジョブのセット、またはデータを使用するクエリを提供できる特定のアカウントのセットを許可する [分析テンプレート](#)。
- ユーザー識別の試みから保護するための [AWS Clean Rooms 差分プライバシー](#)。
- Amazon Athena や Snowflake などの Non-S3 データソース。

この手順では、AWS Clean Rooms コンソールのガイド付きフローオプションを使用して、カスタム分析ルールを設定済みテーブルに追加するプロセスについて説明します。

カスタム分析ルールをテーブルに追加するには (ガイドフロー)

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[分析ルールを設定] を選択します。
5. ステップ 1: 分析ルールタイプを選択する、分析ルールタイプでカスタムオプションを選択します。
6. [作成方法] で [ガイドフロー] を選択し、[次へ] を選択します。
7. ステップ 2: 分析コントロールを指定するで、直接分析コントロールには、目標に基づいてオプションを選択します。

目標	推奨されるアクション
この設定済みテーブルで実行を許可する前に、各新しい分析を確認する	1. 実行できる分析テンプレートで、分析テンプレートの追加を選択します。 2. ドロップダウンリストから適切なコラボレーションと分析テンプレートを選択します。 3. [次へ] を選択します。
特定の共同作業者に、このテーブルをレビューせずに選択したタイプの分析の実行を許可する	1. 分析タイプで、 a. 任意のクエリを選択して、AWS アカウント 指定した によって作成されたクエリを許可します。 b. 任意のクエリを選択して、AWS アカウント 指定した によって作成されたジョブを許可します。 2. AWS アカウント 分析の作成が許可されている で、追加 AWS アカウント を選択します。 3. を入力する AWS アカウント か、ドロップダウンリストから AWS アカウント ID を選択します。 4. (オプション) 別の AWS アカウントを追加する を選択して、別の を追加します AWS アカウント。

目標	推奨されるアクション
	5. [次へ] を選択します。

8. ステップ 3: 分析結果コントロールを指定する

- a. ジョブ結果コントロールでは、追加の結果コントロールはサポートされていないことに注意してください。
- b. クエリ結果コントロールで、出力で許可されていない列については、目標に基づいてクエリ出力で許可する列を選択します。

目標	推奨されるアクション
クエリ出力ですべての列を返すことを許可する	- なしを選択する - 出力に適用される追加の分析に進みます。
クエリ出力で特定の列を返すことを許可しない	- カスタムリストを選択する [許可されていない列を指定] で、クエリ出力から削除する列を選択します。

- c. 出力に適用される追加分析では、目標に基づいて追加の分析をクエリ出力に適用できるかどうかを選択します。

目標	推奨されるオプション
- このテーブルで直接クエリのみを許可します。 - クエリ結果に対する追加分析の実行を拒否します。 - テーブルは直接クエリにのみ使用できません。	許可されていません
このテーブルで直接クエリと追加分析の両方を許可しますが、必須にはしません。	許可

目標	推奨されるオプション
- テーブルは、必要な追加分析のいずれかで処理される直接クエリでのみ使用できることを要求します。 - このテーブルの直接クエリは、返す前にさらに処理する必要があります。	必須

d. [次へ] を選択します。

9. (オプション) ステップ 4: 差分プライバシーを設定する で、差分プライバシーを有効または無効にするかどうかを決定します。

差分プライバシーは、再識別攻撃からデータを保護するための、数学的に証明された手法です。

 Note

AWS Clean Rooms 差分プライバシーは、データが Amazon S3 に保存されているコラボレーションでのみ使用できます。

差分プライバシー では、目的に基づいて差分プライバシーを有効または無効にするかどうかを選択します。

目標	推奨されるアクション
- 再識別の試みに対する保護は必要ありません - テーブルにユーザーレベルのデータがない	[無効にする] を選択します。 [次へ] を選択します。
- 再識別の試みに対する保護が必要です - テーブルにユーザーレベルのデータがある	[オンにする] を選択します。 - プライバシーを保護する 列など、ユーザーの一意の識別子を含むユーザー識別子user_id列を選択します。 コラボレーション内の 2 つ以上のテーブルで差分プライバシーを有効にするには、両方の分析ルールで同じ列を [ユーザー識

目標	推奨されるアクション
	別子列] として設定して、テーブル間でユーザーの定義の一貫性を保つ必要があります。設定に誤りがある場合、クエリを実行できるメンバーには、クエリの実行中にユーザー投稿数 (ユーザーによる広告インプレッション数など) を計算するために選択できる列が 2 つあるというエラーメッセージが表示されます。 3. [次へ] を選択します。

10. [ステップ 5: 確認して設定] で、前のステップで行った選択内容を確認し、必要に応じて編集して、[分析ルールを設定] を選択します。

カスタム分析ルールがテーブルに正常に設定されたことを示す確認メッセージが表示されます。

テーブルへの分析ルールの追加 (JSON エディタ)

次の手順は、AWS Clean Rooms コンソールの JSON エディタオプションを使用してテーブルに分析ルールを追加する方法を示しています。

Note

non-S3データソースを使用する設定済みテーブルは、[カスタム分析ルール](#)のみをサポートします。

集計、リスト、またはカスタム分析ルールをテーブルに追加するには (JSON エディタ)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[分析ルールを設定] を選択します。
5. ステップ 1: 分析ルールタイプを選択し、分析ルールタイプで集計、リスト、またはカスタムオプションを選択します。

6. [作成方法] で [JSON エディタ] を選択し、[次へ] を選択します。
7. [ステップ 2: コントロールを指定] で、クエリ構造を挿入するか ([テンプレートを挿入])、ファイルを挿入するか ([ファイルからインポート]) を選択します。

選択内容	操作
テンプレートを挿入	1. [分析ルール定義] で、選択した分析ルールのパラメータを指定します。 2. Ctrl + スペースバー を押すとオートコンプリートが有効になります。 集計分析ルールパラメータの詳細については、「集計分析ルール - クエリコントロール」を参照してください。 リスト分析パラメータの詳細については、「リスト分析ルール - クエリコントロール」を参照してください。
ファイルからインポート	1. ローカルドライブから JSON ファイルを選択します。 2. 開く をクリックします。 アップロードされたファイルの分析ルールが [分析ルール定義] に表示されます。

8. [次へ] を選択します。
9. [ステップ 3: 確認して設定] で、前のステップで行った選択内容を確認し、必要に応じて編集して、[分析ルールを設定] を選択します。

分析ルールがテーブルに正常に設定されたことを示す確認メッセージが表示されます。

次の手順

設定済みテーブルに分析ルールを設定したら、次の作業に進むことができます。

- [設定済みテーブルをコラボレーションに関連付ける](#)

- (クエリを行えるメンバーとして) [データテーブルにクエリを実行する](#)

設定済みテーブルのコラボレーションへの関連付け

設定済みテーブルを作成して分析ルールを追加したら、そのテーブルをコラボレーションに関連付け、AWS Glue テーブルにアクセスするための AWS Clean Rooms サービスロールを付与できます。

Note

このサービスロールにはテーブルに対するアクセス許可があります。このサービスロールは、クエリを行えるメンバーに代わって許可されたクエリを実行するためにのみ、AWS Clean Rooms によって引き受け可能です。コラボレーションメンバー (データの所有者以外) は誰も、コラボレーションの基になるテーブルにはアクセスできません。データの所有者は差分プライバシーをオンにして、他のメンバーがテーブルのクエリを利用できるようにすることが可能です。

データアクセス予算

設定済みテーブルを関連付けると、データアクセス予算を適用できます。データアクセス予算は、コラボレーション内のクエリ、ジョブ、ML 入力チャネルにテーブルを使用できる回数を制御します。これらの予算は、テーブルの使用を制限することで、組織がリソース使用率を管理し、コストを制御するのに役立ちます。

テーブルがクエリ、ジョブ、または ML 入力チャネルで使用されるたびに、そのテーブルの予算が 1 つ削減されます。予算がゼロに達すると、テーブルを SQL クエリ、Pyspark ジョブ、またはテーブルから派生した ML 入力チャネルの一部として使用することはできません。

定期的に更新される期間ごとの予算、全体的な使用量のライフタイム予算、またはその両方を設定できます。デフォルトでは、テーブルの使用は無制限です。

- 期間ごとの予算 – 指定した期間内にこのテーブルを使用できる回数を制限する更新可能な割り当て。期間は、毎日、毎週、または毎月を設定できます。この予算は、毎日、毎週、または毎月自動的に更新するように設定できます。
- ライフタイム予算 – このテーブルを使用できる合計時間を制限する実行中の割り当て。

設定済みテーブルを関連付ける

以下のトピックでは、コンソールを使用して、設定済みテーブルを AWS Clean Rooms 関連付け、データアクセス予算をコラボレーションに適用する方法について説明します。

AWS SDKs を使用して設定済みテーブルをコラボレーションに関連付ける方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

ステップ 1: 前提条件を満たす

設定済みテーブルを関連付けるには、次の前提条件を満たす必要があります。

- Amazon S3 フォルダの場所を指す AWS Glue テーブル (単一のファイルではない)
- 暗号化された AWS Glue テーブルの場合:
 - テーブルの AWS Glue 復号に AWS KMS キーを使用するアクセス許可を持つサービスロール
 - AWS KMS暗号化された Amazon S3 データセットの場合: サービスロールには、AWS KMS キーを使用して Amazon S3 データを復号するためのアクセス許可も必要です。

暗号化の設定の詳細については、AWS Glue デベロッパーガイドの「[での暗号化の設定 AWS Glue](#)」を参照してください。

AWS Glue テーブルの場所を確認するには:

1. <https://console.aws.amazon.com/glue/> で AWS Glue コンソールを開きます。
2. テーブルの詳細を表示し、場所が S3 フォルダを指していることを確認する

ステップ 2: 設定済みテーブルを関連付ける

設定済みテーブルを関連付けるには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. テーブルを関連付ける方法を選択します。
 - a. 設定済みテーブルの詳細ページから:
 - i. 左のナビゲーションペインで、[テーブル] を選択します。
 - ii. 設定済みテーブルを選択します。

- iii. 設定済みテーブルの詳細ページで、[コラボレーションに関連付ける] を選択します。
 - iv. [テーブルをコラボレーションに関連付ける] ダイアログボックスで、ドロップダウンリストから[コラボレーション] を選択します。
 - b. コラボレーションの詳細ページから:
 - i. 左のナビゲーションペインで、[コラボレーション] を選択します。
 - ii. コラボレーションを選択します。
 - iii. [テーブル] タブで [テーブルに関連付ける] を選択します。
3. テーブルの関連付けページで、次のいずれかを実行します。
 - 既存の設定済みテーブルを選択する – ドロップダウンリストからコラボレーションに関連付ける設定済みテーブル名を選択します。
 - 新しいテーブルの設定 – 新しいテーブルの設定を選択し、新しいテーブルの設定ページのプロンプトに従います。
 - 設定済みテーブルのスキーマと分析ルールを表示する – スキーマと分析ルールの表示を有効にします。
4. [テーブルの関連付けの詳細] で次の操作を行います。
 - a. 関連付けるテーブルの[名前] を入力します。

デフォルトの名前を使用することも、テーブルの名前を変更することもできます。
 - b. (オプション) テーブルの [説明] を入力します。

この説明はクエリの作成に役立ちます。
5. [新しいサービスロールを作成して使用] または [既存のサービスロールを使用] を選択して、[サービスアクセス] 許可を指定します。

 Note

Amazon Athena でバックアップされた設定済みテーブルに関連付ける場合は、ドロップダウンリストから既存のサービスロール名を選択します。サービスロールに IAM と、必要に応じてデータセットへの Lake Formation アクセス許可があることを確認します。

選択内容	結果
新しいサービスロールを作成して使用	• AWS Clean Rooms は、このテーブルに必要なポリシーを持つサービスロールを作成します。 • デフォルトの [サービスロール名] は <code>cleanrooms-<timestamp></code> です。 • ロールを作成してポリシーをアタッチするアクセス許可が必要です。 • 入力データが暗号化されている場合、[このデータは KMS キーで暗号化されます] を選択し、データ入力の復号に使用される AWS KMS key を入力できます。

選択内容	結果
既存のサービスロールを使用	- ドロップダウンリストから [既存のサービスロール名] を選択します。 ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。 ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。 [IAM で表示] 外部リンクを選択してサービスロールを表示します。 既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。 デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。 (オプション) [必要なアクセス許可を備えた事前設定済みポリシーをこのロールに追加] チェックボックスを選択して、必要なアクセス許可をロールにアタッチします。ロールを変更したりポリシーを作成したりするには、アクセス許可が必要です。

 Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、 「」を参照してください [AWS の 管理ポリシー AWS Clean Rooms](#)。
- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに に対する十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。

- ロールポリシーを変更できない場合は、AWS Clean Rooms でサービスロールのポリシーが見つからなかったという内容のエラーメッセージが表示されます。

- 設定済みテーブル関連付けリソースの設定済みテーブル関連付けタグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
- [次へ] を選択します。
- コラボレーション分析ルールの設定ページで、次のいずれかを選択します。
 - はい、今すぐコラボレーション分析ルールを作成する – テーブルをこのコラボレーションに関連付け、コラボレーション分析ルールを作成します
 - いいえ。後でコラボレーション分析ルールを作成します。テーブルをこのコラボレーションにのみ関連付けます。コラボレーション分析ルールは後で作成できます。
- 「はい」を選択した場合は、コラボレーション分析ルールを今すぐ作成します。結果配信では、ド롭ダウンリストからクエリ出力の結果を受信できるメンバーを選択します。
- [次へ] を選択します。
- データアクセス予算の追加ページで、データアクセス予算設定で、次のいずれかを選択します。
 - はい、今すぐデータアクセス予算を追加する – テーブルをこのコラボレーションに関連付け、データアクセス予算を追加します。期間予算、ライフタイム予算、またはその両方を選択できます。
 - いいえ、後でデータアクセス予算を追加します。テーブルをこのコラボレーションにのみ関連付けます。データアクセス予算は後で追加できます。

いいえを選択した場合は、後でデータアクセス予算を追加し、ステップ 15 に進みます。
- はい を選択した場合は、今すぐデータアクセス予算を追加し、次のいずれかの予算設定を選択します。

期間ごとの予算のみ	ライフタイム予算のみ	期間予算とライフタイム予算の両方
1. 期間ごとの追加予算を選択したままにします。 2. 1～1,000,000 の期間ごとの予算額を入力します。	1. 期間ごとの追加予算をクリアします。 2. ライフタイム予算の追加を選択します。	1. 期間ごとの追加予算を選択したままにします。 2. 1～1,000,000 の期間ごとの予算額を入力します。

期間ごとの予算のみ	ライフタイム予算のみ	期間予算とライフタイム予算の両方
3. Period では、Daily、Wkly、または Monthly を選択します。 4. (オプション) 予算の自動更新を毎週選択したままにして、割り当てを更新します。 5. 「ライフタイム予算の追加」をクリアします。	3. 1～1,000,000 のライフタイム予算額を入力します。	3. Period では、Daily、Wkly、または Monthly を選択します。 4. 毎週予算を自動的に更新する を選択したままにします。 5. ライフタイム予算の追加を選択します。 6. 1～1,000,000 のライフタイム予算額を入力します。

13. データアクセス予算の概要で選択内容を確認します。

Example例

たとえば、期間ごとの予算額として 1,000 を選択し、期間を週次に設定し、週次自動更新予算チェックボックスを選択したままにして、ライフタイム予算を 1,000,000 に設定すると、アクセス予算の概要に次のメッセージが表示されます。毎週、このテーブルはクエリまたはジョブの実行に最大 1,000 回使用できます。この予算は毎週日曜日の 00:00 UTC に自動的に更新されるように設定され、このテーブルがライフタイム予算である 1,000,000 回の使用量に達するまで更新され続けます。

14. (オプション) アクセス予算リソースのデータアクセス予算タグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
15. [次へ] を選択します。
16. レビューと作成ページの情報を確認します。
- a. セクションを編集する必要がある場合は、編集 を選択します。
- b. 設定を編集し、次へを選択します。
17. [テーブルを関連付ける] を選択します。

ステップ 3: 次のステップ

設定済みデータテーブルをコラボレーションに関連付けたら、次の作業に進むことができます。

- 設定済みテーブルに[コラボレーション分析ルールを追加する](#)
- コラボレーションクリエイターの場合は[コラボレーションを編集する](#)
- (クエリを行えるメンバーとして) [データテーブルにクエリを実行する](#)

Configuring a data access budget

共同作業者は、データアクセス予算を表示、追加、編集、削除して、ワークフローでテーブルを使用できる回数に制限を設定できます。これらの予算を使用してデータとコストを管理します。

テーブルがクエリされるか、テーブルから派生した ML 入力チャネルを使用してジョブが実行されるたびに、そのテーブルの予算が 1 つ削減されます。予算がゼロに達すると、テーブルをクエリしたり、テーブルから派生した ML 入力チャネルを使用して ML ジョブを実行したりすることはできません。

定期的に更新される期間ごとの予算、全体的な使用量のライフタイム予算、またはその両方を設定できます。デフォルトでは、テーブルの使用は無制限です。

- 期間ごとの予算 – 指定した期間内にこのテーブルを使用できる回数を制限する更新可能な割り当て。期間は、毎日、毎週、または毎月を設定できます。この予算は、毎日、毎週、または毎月自動的に更新するように設定できます。
- ライフタイム予算 – このテーブルを使用できる合計時間を制限する実行中の割り当て。

トピック

- [データアクセス予算の表示](#)
- [既存の関連テーブルへのデータアクセス予算の追加](#)
- [データアクセス予算の編集](#)
- [データアクセス予算の削除](#)

データアクセス予算の表示

データアクセス予算は、テーブルタブまたはテーブルの詳細ページから表示できます。

データアクセス予算を表示するには

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. 次のいずれかを行います。
 - 残りのデータアクセス予算列で、詳細を表示する予算を選択します。
 - テーブルを選択し、テーブルの詳細ページで下にスクロールしてデータアクセス予算の詳細セクションを表示します。

既存の関連テーブルへのデータアクセス予算の追加

コラボレーションメンバーは、既存の関連テーブルにデータアクセス予算を追加できます。

既存の関連付けられたテーブルにデータアクセス予算を追加するには

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. データアクセス予算を追加するテーブルの横にあるオプションボタンを選択します。
6. Actions ドロップダウンリストから、Data Access Budget で、Add (予算がまだない場合) を選択します。
7. 次のいずれかの予算設定を選択します。

期間ごとの予算のみ	ライフタイム予算のみ	期間予算とライフタイム予算の両方
1. 期間ごとの追加予算を選択したままにします。	1. 期間ごとの予算の追加をクリアします。	1. 期間ごとの追加予算を選択したままにします。

期間ごとの予算のみ	ライフタイム予算のみ	期間予算とライフタイム予算の両方
2. 1 ~ 1,000,000 の期間ごとの予算額を入力します。	2. ライフタイム予算の追加を選択します。	2. 1 ~ 1,000,000 の期間ごとの予算額を入力します。
3. Period では、Daily、Wkly、または Monthly を選択します。	3. 1 ~ 1,000,000 のライフタイム予算額を入力します。	3. Period では、Daily、Wkly、または Monthly を選択します。
4. (オプション) 予算の自動更新を毎週選択したままにして、割り当てを更新します。		4. 毎週予算を自動的に更新するを選択したままにします。
5. 「有効期間予算の追加」をクリアします。		5. ライフタイム予算の追加を選択します。
		6. ライフタイム予算額を 1 ~ 1,000,000 の間で入力します。

8. データアクセス予算の概要で選択内容を確認します。

9. Example例

例えば、期間ごとの予算額として 1,000 を選択し、期間を週次に設定し、週次自動更新予算チェックボックスを選択したままにして、ライフタイム予算を 1,000,000 に設定すると、アクセス予算の概要に次のメッセージが表示されます。毎週、このテーブルはクエリまたはジョブの実行に最大 1,000 回使用できます。この予算は毎週日曜日の 00:00 UTC に自動的に更新されるように設定され、このテーブルがライフタイム予算である 1,000,000 回の使用量に達するまで更新され続けます。

10. (オプション) アクセス予算リソースのデータアクセス予算タグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。

11. データアクセス予算の追加を選択します。

データアクセス予算の編集

コラボレーションメンバーは、データアクセス予算を編集できます。データアクセス予算を編集すると、現在の予算残高がリセットされます。

データアクセス予算は、テーブルタブまたはテーブルの詳細ページから編集できます。

Tables tab

Tables タブからデータアクセス予算を編集するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. 編集するテーブルの横にあるオプションボタンを選択します。
6. Actions ドロップダウンリストから、Data Access Budget で Edit を選択します。
7. データアクセス予算の編集ページで、期間ごとの予算またはライフタイム予算情報を更新します。
8. データアクセス予算の概要を表示して、行った編集が正しいことを確認します。
9. [Save changes] (変更の保存) をクリックします。

Table details page

テーブルの詳細ページからデータアクセス予算を編集するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. テーブルを選択します。
6. テーブルの詳細ページで、データアクセス予算の詳細セクションまで下にスクロールします。
7. [アクション] ドロップダウンリストから、[編集] を選択します。
8. データアクセス予算の編集ページで、期間ごとの予算またはライフタイム予算情報を更新します。
9. [Save changes] (変更の保存) をクリックします。

データアクセス予算の削除

データアクセス予算は、テーブルタブまたはテーブルの詳細ページから削除できます。

Tables tab

Tables タブからデータアクセス予算を削除するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. 削除するテーブルの横にあるオプションボタンを選択します。
6. Actions ドロップダウンリストから、Data Access Budget で Delete を選択します。



Important

このアクションを元に戻すことはできません。データアクセス予算は無制限にリセットされます。

7. データアクセス予算を削除することが確実な場合は、削除を選択します。

Table details page

テーブルの詳細ページからデータアクセス予算を削除するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. テーブルを選択します。
6. テーブルの詳細ページで、データアクセス予算の詳細セクションまでスクロールダウンします。
7. [アクション] ドロップダウンリストから、[削除] を選択します。

 Important

このアクションを元に戻すことはできません。データアクセス予算は無制限にリセットされます。

8. データアクセス予算を削除することが確実な場合は、削除を選択します。

設定済みテーブルへのコラボレーション分析ルールの追加

コラボレーション分析ルールを使用すると、このコラボレーションに固有のコントロールを指定できます。これらのコントロールは、設定済みテーブル分析ルールと連携して、このコラボレーション内でこのテーブルを分析する方法を決定します。

[設定済みテーブルを作成し](#)、[分析ルールを追加し](#)、[それをコラボレーションに関連付け](#)たら、設定済みテーブルにコラボレーション分析ルールを追加します。テーブルが直接分析をサポートするように設定されている場合、または追加分析を許可するように設定されている場合は、コラボレーション分析ルールを追加する必要があります。

- 直接分析 – テーブルは、それを直接分析するクエリで使用できます。例えば、集計測定分析またはアクティベーション用の ID のリストを出力するクエリの場合です。
- 追加分析 – テーブルは、それを直接分析するクエリに加えて、追加分析への入力としても使用できます。たとえば、テーブルは類似 ML モデルのシードであるクエリや、カスタム ML モデルの ML 入力チャネルで使用できます。

コラボレーション分析ルールをテーブルに追加するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テーブル] タブの [自身が関連付けたテーブル] で、コラボレーションに関連付けた設定済みテーブルを表示します。
 - [直接分析のステータス] または [追加分析のステータス] のステータスが [準備ができています] の場合は、そのテーブルに対してクエリを実行できます。

- [直接分析のステータス] または [追加分析のステータス] のステータスが [準備ができていません] の場合は、ステータスを選択し、ダイアログボックスで [設定] を選択します。
5. [コラボレーション分析ルールを設定] ページで、[設定済みテーブル分析ルールを表示] を展開して詳細を表示します。
 6. [許可されている追加分析] で、目標に応じてオプションを選択します。

目標	推奨されるオプション
テーブルでの追加分析を許可します。	すべて
特定のメンバーによるテーブルでの追加分析のみを許可します。	特定のメンバーによるあらゆるもの
テーブルでの特定分析のみを許可します。	カスタムリスト

7. [結果の配信] で、[クエリ出力の結果を受け取ることができるメンバー] ドロップダウンから、結果を受け取ることができるユーザーを指定します。
8. [分析ルールを設定] を選択します。

差分プライバシーポリシーの設定 (オプション)

Note

AWS Clean Rooms 差分プライバシーは、データが Amazon S3 に保存されているコラボレーションでのみ使用できます。

この手順では、AWS Clean Rooms コンソールのガイド付きフローオプションを使用して、コラボレーションで差分プライバシーポリシーを設定するプロセスについて説明します。これは、差分プライバシー保護のあるすべてのテーブルに対する 1 回限りの手順です。

差分プライバシー設定を設定するには (ガイドフロー)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。

4. コラボレーションページの [テーブル] タブで、[差分プライバシーポリシーを設定] を選択します。
5. [差分プライバシーポリシーを設定] ページで、次のプロパティの値を選択します。
 - プライバシー予算
 - プライバシー予算を毎月更新
 - クエリごとに追加されるノイズ

デフォルト値を使用するか、特定のユースケースをサポートするカスタム値を入力できます。[プライバシー予算] と [クエリごとに追加されるノイズ] の値を選択した後、データに対するすべてのクエリ全体で使用可能な集計の数に関して、結果のユーティリティをプレビューできます。

6. [設定] を選択します。

コラボレーションの差分プライバシーポリシーが正常に設定されたことを示す確認メッセージが表示されます。

差分プライバシーが設定され、次の作業に進むことができます。

- (クエリを行えるメンバーとして) [データテーブルにクエリを実行する](#)
- [コラボレーション](#) (コラボレーションクリエーターの場合)

差分プライバシー使用状況ログの表示

差分プライバシーでデータを保護しているコラボレーションメンバーは、差分プライバシーでコラボレーションを作成した後で、プライバシー予算の使用状況をモニタリングできます。

集計が行われた回数と使用されたプライバシー予算の金額を確認するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [Tables] (テーブル) タブを選択します。
5. [使用状況ログを表示] (青色のテキスト) を選択します。

6. プライバシー予算や提供したユーティリティの提供量など、使用状況の詳細を表示します。

差分プライバシーポリシーの編集

差分プライバシーポリシーを設定した後は、プライバシーのニーズをより適切に反映するようにいつでも更新できます。

差分プライバシーポリシーを編集するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. コラボレーションページの [テーブル] タブの [自分が関連付けたテーブル] で、[編集] を選択します。
5. [差分プライバシーの編集] ページで、次のプロパティの新しい値を選択します。
 - [プライバシー予算] — コラボレーション中の任意の時点で、スライダーバーを動かして予算を増減できます。クエリを実行できるメンバーがデータのクエリを開始した後は、予算を減らすことはできません。プライバシー予算を増やす AWS Clean Rooms と、は新しく追加されたプライバシー予算を使用する前に、完全に消費されるまで既存の予算を引き続き使用します。
 - [クエリごとに追加されるノイズ] — コラボレーション中の任意の時点で、スライダーバーを動かして [クエリごとに追加されるノイズ] を増減できます。

Note

[インタラクティブサンプル] を選択して、[プライバシー予算] と [クエリごとに追加されるノイズ] の値が異なると、実行できる集計関数の数にどのように影響するかを調べることができます。

[プライバシー予算の更新] の値は変更できません。選択を変更するには、差分プライバシーポリシーを削除し、新しく作成する必要があります。

6. [Save changes] (変更の保存) をクリックします。

差分プライバシーポリシーが正常に編集されたことを示す確認メッセージが表示されます。

差分プライバシーポリシーの削除

差分プライバシーポリシーはコラボレーションの [テーブル] タブから削除できます。

差分プライバシーポリシーを削除するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. コラボレーションページの [テーブル] タブで、[差分プライバシーポリシー] の横にある [削除] を選択します。
5. 差分プライバシーポリシーを削除してよい場合は、[削除] を選択します。

差分プライバシーポリシーを削除すると、そのポリシーのプライバシー予算使用状況ログにアクセスできなくなります。差分プライバシーポリシーが削除されると、差分プライバシーがオンになっているテーブルをクエリできなくなります。

計算された差分プライバシーパラメータの表示

差分プライバシーに関する専門知識を持つユーザーは、コラボレーションの分析タブから計算された差分プライバシーパラメータを表示できます。

計算された差分プライバシーパラメータを表示するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. 分析タブの結果セクションで、計算された差分プライバシーパラメータの表示を選択します。

[計算された差分プライバシーパラメータ] のテーブルでは、集計関数の感度値が表示されます。これは、1 人のユーザーのレコードが追加、削除、または変更された場合に関数の結果が変化する可能性のある最大値として定義されます。リストには、次の差分プライバシーパラメータが含まれます。

- [ユーザー寄与度上限] (UCL) は、SQL クエリでユーザーが入力する行の最大数です。例えば、各ユーザーが複数のインプレッションを持つことができる指定されたキャンペーンで一致するインプレッションの合計数をカウントする場合、AWS Clean Rooms 差分プライバシー計算が正確になるように、差分プライバシーは 1 人のユーザーのインプレッション数をバインドする必要があります。つまり、ユーザーが境界よりも多くのインプレッションを持つ場合、は計算された UCL 値に従ってそのユーザーのインプレッションの均一なランダムサンプル AWS Clean Rooms を自動的に取得し、クエリの実行中にそのユーザーの残りのインプレッションを除外します。一意のユーザー数をカウントする場合、UCL 値は 1 に等しくなります。これは、1 人のユーザーを追加、削除、または変更することで、個々のユーザーのカウントが最大で 1 だけ変更される可能性があるためです。
- [最小値] は、`sum()` のような集計関数内で使用される式の下限值です。例えば、式が `purchase_value` という列の場合、最小値はその列の下限值です。
- [最大値] は、`sum()` のような集計関数内で使用される式の上限值です。例えば、式が `purchase_value` という列の場合、最大値はその列の上限值です。

[計算された差分プライバシーパラメータ] のテーブルでは、これらのパラメータを使用してクエリ結果のノイズの総量の理解を深めることができます。たとえば、クエリごとに追加された設定済みのノイズが 30 ユーザーで、`COUNT DISTINCT (user_id)` クエリが実行されると、の感度が 1 であるため、AWS Clean Rooms 差分プライバシーは -30 から 30 の間のランダムノイズを高い確率で追加 `COUNT DISTINCT` します。同じ設定の `COUNT` クエリの場合、AWS Clean Rooms 差分プライバシーは、1 人のユーザーがクエリ結果に複数の行を寄稿する可能性があるため、ユーザーの寄与度上限に応じてスケールされる統計的ノイズを追加します。`SUM (purchase_value)` すべての列値が正であるような `SUM` クエリの場合、合計ノイズはユーザー寄与制限に最大値を掛けてスケールされます。AWS Clean Rooms 差分プライバシーは、クエリの実行時にノイズの追加を実行するために感度パラメータを自動的に計算し、プライバシー予算を枯渇させます。感度パラメータはデータに依存するため、プライバシー予算を使い果たす必要があります。

テーブルと分析ルールの表示

コラボレーションに関連付けられたテーブルと分析ルールを表示するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。

4. [Tables] (テーブル) タブを選択します。
5. 次のいずれかを選択します。
 - a. コラボレーションに関連付けられている自分のテーブルを表示するには、[自分が関連付けたテーブル] でテーブル (青色のテキスト) を選択します。
 - b. コラボレーションに関連付けられている他のテーブルを表示するには、[コラボレーターが関連付けたテーブル] でテーブル (青色のテキスト) を選択します。
6. テーブルの詳細と分析ルールが、テーブルの詳細ページに表示されます。

設定済みテーブルの編集

前提条件:

- にアクセスできる AWS アカウント 。 AWS Clean Rooms

以下のセクションでは、Amazon S3、Amazon Athena、Snowflake データソースのテーブルの名前、説明、設定の詳細を編集する方法について説明します。

AWS SDKs を使用して設定済みテーブルを編集する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

設定済みテーブルを編集するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 自分が作成した設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、**編集** を選択します。
5. 設定を編集します。
6. [Save changes] (変更の保存) をクリックします。

設定済みテーブルのタグの編集

コラボレーションメンバーは、設定済みテーブルの作成後に、[設定済みのテーブル] タブで設定済みテーブルリソースのタグを管理できます。

設定済みテーブルのタグを編集するには

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 自分が作成した設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[タグ] セクションまで下方向にスクロールします。
5. [Manage tags (タグの管理)] を選択します。
6. [Manage tags] (タグの管理) ページで、次の操作を実行できます。
 - タグを削除するには[削除] を選択してください。
 - タグを追加するには、[新しいタグの追加] を選択します。
 - 変更を保存するには、変更の保存を選択します。

設定済みテーブル分析ルールの編集

設定済みテーブルの分析ルールを編集するには

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 自分が作成した設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[集計分析ルール]、[分析ルールを一覧表示]、または [カスタム分析ルール] セクションまで下方向にスクロールします (どれを選択するかは、設定済みテーブルにどのタイプの分析ルールを選択したかによって異なります)。
5. [編集] を選択します。
6. [分析ルールを編集] ページで、次の操作を実行できます。
 - 次の方法で [分析ルール定義] を変更します。
 - JSON エディタを変更します。
 - [ファイルからインポート] を選択して、新しい分析ルール定義をアップロードします。
 - 以下のオプションから選択して、コラボレーションでメンバーに表示される内容をプレビューします。
 - テーブルビュー

- JSON
- クエリの例

7. [変更を保存] を選択して、変更を保存します。

設定済みテーブル分析ルールの削除

Warning

このアクションは元に戻すことができず、関連するすべてのリソースに影響します。

設定済みテーブルの分析ルールを削除するには

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. 自分が作成した設定済みテーブルを選択します。
4. 設定済みテーブルの詳細ページで、[集計分析ルール]、[分析ルールを一覧表示]、または [カスタム分析ルール] セクションまで下方向にスクロールします (どれを選択するかは、設定済みテーブルにどのタイプの分析ルールを選択したかによって異なります)。
5. [削除] を選択します。
6. 分析ルールを削除してよい場合は、[削除] を選択します。

設定済みテーブルの許可されていない列

許可されていない出力列の設定は、AWS Clean Rooms カスタム分析ルールのコントロールであり、クエリ結果に投影できない列のリスト (存在する場合) を定義できます。このリストで参照されている列は、「許可されていない出力列」と見なされます。つまり、変換、エイリアシング、またはその他の手段を通じたこのような列への参照は、クエリの最終 SELECT (射影) に存在しない可能性があります。

この機能では、列が出力に直接射影されることを禁止していますが、基になる値が他のメカニズムを介して間接的に推測されることを完全に防ぐわけではありません。これらの列は、最終射影で参照されていない限り、射影句 (サブクエリや共通テーブル式 (CTE) など) で引き続き使用できます。

許可されていない出力列の設定により、ユースケースと対応するプライバシー要件に基づく分析テンプレートレベルのレビューと組み合わせて、テーブルにコントロールを適用およびコード化する柔軟性が得られます。

この設定方法に関する詳細については、「[カスタム分析ルール of テーブルへの追加 \(ガイドフロー\)](#)」を参照してください。

例

次の例は、許可されていない出力列コントロールがどのように適用されるかを示しています。

- メンバー A は、メンバー B とコラボレーションしています。
- メンバー B は、クエリを実行できるメンバーです。
- メンバー A は、age、gender、email、および name の各列を持つ users テーブルを定義します。age および name 列は、許可されていない出力列です。
- メンバー B は、age、gender、および owner_name 列の類似したセットを持つ pets テーブルを定義します。ただし、出力列に制約は設定しません。このため、テーブル内のすべての列をクエリで自由に射影できます。

メンバー B が次のクエリを実行すると、ブロックされます。これは、許可されていない出力列を直接射影できないためです。

```
SELECT
  age
FROM
  users
```

メンバー B が次のクエリを実行すると、ブロックされます。これは、許可されていない出力列をプロジェクトスターを介して暗黙的に射影できないためです。

```
SELECT
  *
FROM
  users
```

メンバー B が次のクエリを実行すると、ブロックされます。これは、許可されていない出力列の変換を射影できないためです。

```
SELECT
  COUNT(age)
FROM
  users
```

メンバー B が次のクエリを実行すると、ブロックされます。これは、許可されていない出力列はエイリアスを使用して最終射影で参照できないためです。

```
SELECT
  count_age
FROM
  (SELECT COUNT(age) AS count_age FROM users)
```

メンバー B が次のクエリを実行すると、ブロックされます。これは、変換された制限付き列が出力に射影されるためです。

```
SELECT
  CONCAT(name, email)
FROM
  users
```

メンバー B が次のクエリを実行すると、ブロックされます。これは、CTE で定義された許可されていない出力列を最終射影で参照できないためです。

```
WITH cte AS (
  SELECT
    age AS age_alias
  FROM
    users
)
SELECT age_alias FROM cte
```

メンバー B が次のクエリを実行すると、ブロックされます。これは、許可されていない出力列を最終射影のソートキーまたはパーティションキーとして使用できないためです。

```
SELECT
  LISTAGG(gender) WITHIN GROUP (ORDER BY age) OVER (PARTITION BY age)
```

```
FROM
  users
```

メンバー B が次のクエリを実行すると、成功します。これは、許可されていない出力列の一部である列は、結合句やフィルター句など、クエリ内の他のコンストラクト間で引き続き使用できるためです。

```
SELECT
  u.name,
  p.gender,
  p.age
FROM
  users AS u
JOIN
  pets AS p
ON
  u.name = p.owner_name
```

同じシナリオで、メンバー B はフィルターまたはソートキーとして users の name 列を使用することもできます。

```
SELECT
  u.email,
  u.gender
FROM
  users AS u
WHERE
  u.name = 'Mike'
ORDER BY
  u.name
```

さらに、users からの許可されていない出力列は、次のようなサブクエリや CTE などの中間射影で使用できます。

```
WITH cte AS (
  SELECT
    u.gender,
    u.id,
```



```
    u.first_name
FROM
    users AS u
)
SELECT
    first_name
FROM
    (SELECT cte.gender, cte.id, cte.first_name FROM cte)
```

設定済みテーブルの関連付けの編集

コラボレーションメンバーは、作成した設定済みテーブルの関連付けを編集できます。

設定済みテーブルの関連付けを編集するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テーブル] タブを選択します。
5. [自分が関連付けたテーブル] で、テーブルを選択します。
6. テーブルの詳細ページで、[テーブルの関連付けの詳細] まで下方向にスクロールします。
7. [編集] を選択します。
8. [設定済みのテーブル関連付けを編集] ページで、[説明] または [サービスアクセス情報] を更新します。
9. [Save changes] (変更の保存) をクリックします。

設定済みテーブルの関連付けの解除

コラボレーションメンバーは、設定済みテーブルとコラボレーションの関連付けを解除できます。このアクションにより、クエリを行えるメンバーはテーブルでクエリを実行できなくなります。

設定済みテーブルの関連付けを解除するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

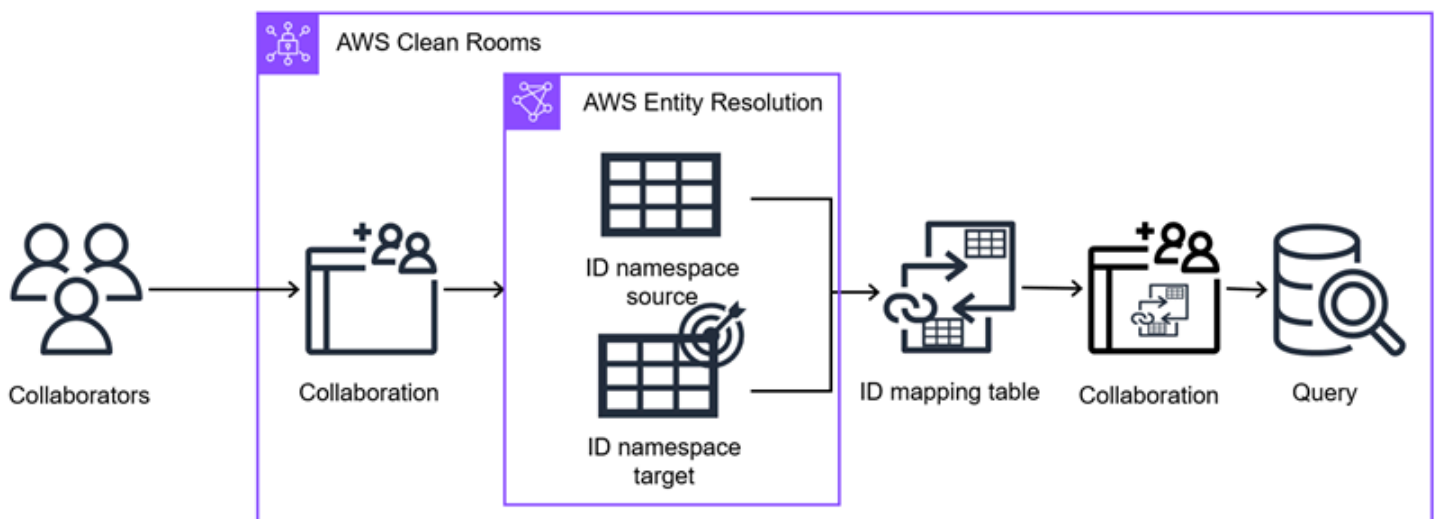
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テーブル] タブを選択します。
5. [自分が関連付けたテーブル] で、関連付けを解除するテーブルの横にあるオプションボタンを選択します。
6. [関連付け解除] を選択してください。
7. ダイアログボックスで [関連付け解除] を選択すると、設定済みテーブルの関連付けの解除が確定し、クエリを行えるメンバーがテーブルでクエリを実行できなくなります。

AWS Entity Resolution in AWS Clean Rooms

AWS Entity Resolution で AWS Clean Rooms、ソースからターゲットにデータを変換し、変換されたデータを ID マッピングテーブルに入力して、データをクエリできます。

まず、コラボレーションを作成し AWS Clean Rooms、招待 AWS アカウント する を追加するか、メンバーシップを作成して招待されたコラボレーションに参加します。次に、2 つのデータテーブルで ID マッピングを実行します。これを行うには、既存の ID 名前空間ソースを関連付けるか、AWS Entity Resolution で新しい ID 名前空間ソースを作成します。コラボレーションの他のメンバーは、既存の ID 名前空間ターゲットを関連付けるか、新しい ID 名前空間ターゲットを作成します。次に、2 つの関連付けられた ID 名前空間から ID マッピングテーブルを作成して入力します。最後に、クエリを実行できるメンバーは、ID マッピングテーブルで結合することで、2 つのデータテーブル間でクエリを実行します。

次の図は、AWS Entity Resolution で を使用する方法をまとめたものです AWS Clean Rooms。



Note

現在サポートされているトランスコードサービスプロバイダーは LiveRamp です。LiveRamp は、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 (オレゴン) で利用できます。

トピック

- [の ID 名前空間 AWS Clean Rooms](#)
- [の ID マッピングテーブル AWS Clean Rooms](#)

の ID 名前空間 AWS Clean Rooms

ID 名前空間は、ID テーブルを囲むラッパーで、データセットとその使用方法を ID マッピングワークフローで説明するメタデータを提供できます。ID マッピングワークフローは、指定された ID マッピング方法に基づいて、入力データソースから入力データターゲットにデータをマッピングするデータ処理ジョブです。これにより、ID マッピングテーブルが生成されます。

ID 名前空間には、[ソース] と [ターゲット] の 2 種類があります。[ソース] には、ID マッピングワークフローで処理されるソースデータの設定が含まれています。[ターゲット] には、すべてのソースが解決するターゲットデータの設定が含まれています。2 つの間で解決する入力データを定義するには AWS アカウント、ID 名前空間ソースと ID 名前空間ターゲットを作成して、データを 1 つのセット (ソース) から別のセット (ターゲット) に変換します。

新しい ID 名前空間を作成するか、既存の名前空間を関連付けることができます。で ID 名前空間を作成する方法の詳細については AWS Entity Resolution、AWS Entity Resolution 「ユーザーガイド」の「[ID 名前空間を使用して入力データを定義する](#)」を参照してください。

トピック

- [新しい ID 名前空間の作成と関連付け](#)
- [既存の ID 名前空間の関連付け](#)
- [ID 名前空間の関連付けの編集](#)
- [ID 名前空間の関連付け解除](#)

新しい ID 名前空間の作成と関連付け

コラボレーションの各メンバーは、ID データのクエリを実行する ID マッピングテーブルを作成する前に、ID 名前空間の [ソース] または [ターゲット] のいずれかを作成して関連付ける必要があります。

で ID 名前空間を既に作成している場合は AWS Entity Resolution、「」に進みます [既存の ID 名前空間の関連付け](#)。

新しい ID 名前空間を作成して関連付けるには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。

3. コラボレーションを選択します。
4. [エンティティ解決] タブで、[ID 名前空間を関連付ける] を選択します。
5. [ID 名前空間を関連付ける] ページで、[エンティティ解決データ] の [ID 名前空間を作成] を選択します。

AWS Entity Resolution コンソールが新しいタブに表示されます。

6. AWS Entity Resolution コンソールの [ID 名前空間を作成] ページのプロンプトに従います。
 - a. [詳細] で、[ID 名前空間名]、[説明] を入力し、[ID 名前空間タイプ] ([ソース] または [ターゲット]) を選択します。
 - b. [ID 名前空間メソッド] で、ルールベースのマッチングに対して [ルールベース] のメソッドを選択するか、サードパーティーのトランスコーディングに対して [プロバイダーサービス] を選択します。
 - c. 選択した ID 名前空間メソッドに応じて、[データ入力タイプ] を指定します。
 - d. [ID 名前空間の作成] を選択します。
7. AWS Clean Rooms コンソールに戻ります。
8. [エンティティ解決データ] の [ID 名前空間を関連付ける] ページで、コラボレーションに関連付ける [AWS Entity Resolution ID 名前空間] のソースまたはターゲットをドロップダウンリストから選択します。
9. [関連付けの詳細] で、次の手順を実行します。
 - a. 関連付けられた ID 名前空間の [名前] を入力します。

デフォルトの名前を使用することも、この ID 名前空間の名前を変更することもできます。

- b. (オプション) ID 名前空間の [説明] を入力します。

この説明はクエリの作成に役立ちます。

10. オプションを選択し、推奨されるアクションを実行して、[AWS Clean Rooms アクセス] 許可を指定します。

オプション	推奨されるアクション
アクセス許可ポリシーの追加と管理を AWS Clean Rooms に許可する	AWS Clean Rooms は、この関連付けに必要なポリシーを持つサービスロールを作成します。

オプション	推奨されるアクション
許可を手動で追加して管理	次のいずれかを行います。 • [リソースポリシー] を確認して、必要なアクセス許可をポリシーに追加します。 • [ポリシーステートメントを追加] を選択して、既存のポリシーを使用します。 ロールを変更したりポリシーを作成したりするには、アクセス許可が必要です。  Note ロールポリシーを変更できない場合は、<code>aws:iam:policy:GetPolicy</code> を見つけ AWS Clean Rooms からのエラーメッセージが表示されます。

11. (オプション) [高度な ID マッピングテーブルの設定] で、ID 名前空間から取得される列のデフォルトの保護を変更します。

ID マッピングテーブルは、[sourceID] 列と [targetID] 列の両方で INNER JOIN のみを許可するようにデフォルトで設定されています。この設定を変更して、この ID 名前空間 (sourceID または targetID) からの列をクエリ内の任意の場所で許可できます。

目標	推奨されるオプション
列を「結合列」として分類し、INNER JOIN 句でのみ許可する	あり
列を「ディメンション列」として分類し、クエリの JOIN 句、SELECT、WHERE、および GROUP BY ステートメントを含めて、クエリ内のどの場所でも許可します。	いいえ。クエリのどの場所でも許可します

12. (オプション) ID 名前空間リソースで [タグ] を有効にする場合は、[新しいタグを追加] を選択し、[キー] と [値] のペアを入力します。
13. 関連付ける を選択してください。
14. [エンティティ解決] タブの [関連付けられた ID 名前空間] テーブルで、関連付けられた ID 名前空間を表示し、ID 名前空間タイプが正しいことを確認します ([ソース] または [ターゲット])。

コラボレーションのすべてのメンバーが ID 名前空間を関連付けたら、[ID マッピングテーブルを作成](#)してデータのクエリを実行できます。

既存の ID 名前空間の関連付け

この手順では、各メンバーはコラボレーションで既存の ID 名前空間ソースまたは ID 名前空間ターゲットを関連付けます。

既存の ID 名前空間を関連付けるには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [エンティティ解決] タブで、[ID 名前空間を関連付ける] を選択します。
5. [エンティティ解決データ] の [ID 名前空間を関連付ける] ページで、コラボレーションに関連付ける [AWS Entity Resolution ID 名前空間] のソースまたはターゲットをドロップダウンリストから選択します。
6. [関連付けの詳細] で、次の手順を実行します。

- a. 関連付けられた ID 名前空間の [名前] を入力します。

デフォルトの名前を使用することも、この ID 名前空間の名前を変更することもできます。

- b. (オプション) ID 名前空間の [説明] を入力します。

この説明はクエリの作成に役立ちます。

7. オプションを選択し、推奨されるアクションを実行して、[AWS Clean Rooms アクセス] 許可を指定します。

オプション	推奨されるアクション
アクセス許可ポリシーの追加と管理を AWS Clean Rooms に許可する	AWS Clean Rooms は、この関連付けに必要なポリシーを持つサービスロールを作成します。
許可を手動で追加して管理	次のいずれかを行います。 • [リソースポリシー] を確認して、必要なアクセス許可をポリシーに追加します。 • [ポリシーステートメントを追加] を選択して、既存のポリシーを使用します。 ロールを変更したりポリシーを作成したりするには、アクセス許可が必要です。  Note ロールポリシーを変更できない場合は、サービスロールのポリシーを見つけ AWS Clean Rooms になかったことを示すエラーメッセージが表示されます。

8. (オプション) [高度な ID マッピングテーブルの設定] で、ID 名前空間から取得される列のデフォルトの保護を変更します。

ID マッピングテーブルは、[sourceID] 列と [targetID] 列の両方で INNER JOIN のみを許可するようにデフォルトで設定されています。この設定を変更して、この ID 名前空間 (sourceID または targetID) からの列をクエリ内の任意の場所で許可できます。

目標	推奨されるオプション
列を「結合列」として分類し、INNER JOIN 句でのみ許可します。	あり

目標	推奨されるオプション
列を「ディメンション列」として分類し、クエリの JOIN 句、SELECT、WHERE、および GROUP BY ステートメントを含めて、クエリ内のどの場所でも許可します。	いいえ。クエリのどの場所でも許可します

9. (オプション) ID 名前空間リソースで [タグ] を有効にする場合は、[新しいタグを追加] を選択し、[キー] と [値] のペアを入力します。
10. 関連付ける を選択してください。
11. [エンティティ解決] タブの [関連付けられた ID 名前空間] テーブルで、関連付けられた ID 名前空間を表示し、ID 名前空間タイプが正しいことを確認します ([ソース] または [ターゲット])。

コラボレーションのすべてのメンバーが ID 名前空間を関連付けたら、[ID マッピングテーブルを作成](#)してデータのクエリを実行できます。

ID 名前空間の関連付けの編集

コラボレーションメンバーは、作成した ID 名前空間の関連付けを編集できます。

ID 名前空間の関連付けを編集するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [エンティティ解決] タブを選択します。
5. [関連付けられた ID 名前空間] で、ID 名前空間を選択します。
6. ID 名前空間の詳細ページで、下にスクロールして [ID 名前空間の関連付けの詳細] を表示します。
7. [編集] を選択します。
8. [ID 名前空間の関連付けを編集] ページで、次のいずれかを編集します。
 - a. [関連付けの詳細] で、[名前] または [説明] を更新します。

- b. (オプション) [高度な ID マッピングテーブルの設定] で、ID 名前空間から取得される列のデフォルトの保護を変更します。

ID マッピングテーブルは、[sourceID] 列と [targetID] 列の両方で INNER JOIN のみを許可するようにデフォルトで設定されています。この設定を変更して、この ID 名前空間 (sourceID または targetID) からの列をクエリ内の任意の場所で許可できます。

目標	推奨されるオプション
列を「結合列」として分類し、INNER JOIN 句でのみ許可する	あり
列を「ディメンション列」として分類し、クエリの JOIN 句、SELECT、WHERE、および GROUP BY ステートメントを含めて、クエリ内のどの場所でも許可します。	いいえ。クエリのどの場所でも許可します

9. [Save changes] (変更の保存) をクリックします。

ID 名前空間の関連付け解除

コラボレーションメンバーは、ID 名前空間とコラボレーションの関連付けを解除できます。このアクションにより、クエリを行えるメンバーはテーブルでクエリを実行できなくなります。

Warning

ID 名前空間の関連付けをコラボレーションから解除すると、派生 ID マッピングテーブルからすべてのデータが削除され、クエリを実行できなくなります。

例えば、3 つの異なる ID マッピングテーブルで ID 名前空間の関連付けがソースとして使用された場合、ID 名前空間の関連付けを解除すると、これらの ID マッピングテーブルのすべてのデータが削除されます。

ID 名前空間の関連付けを解除するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。

3. コラボレーションを選択します。
4. [エンティティ解決] タブを選択します。
5. [関連付けられた ID 名前空間] で、関連付けを解除する ID 名前空間の横にあるオプションボタンを選択します。
6. [関連付け解除] を選択してください。
7. ダイアログボックスで、[関連付け解除] を選択して ID 名前空間を切断する決定を確認します。このアクションを行うと、クエリを実行できるメンバーが ID マッピングテーブルにアクセスできなくなります。

コラボレーションのメンバーによって ID 名前空間の 1 つが削除された場合、ソースがコラボレーションから退出すると、ID マッピングテーブルに再度データを入力できなくなります。

ID マッピングテーブルに以前にデータが入力されていた場合であっても、ID 名前空間の関連付けを解除すると、そのテーブルでクエリを実行できなくなります。

の ID マッピングテーブル AWS Clean Rooms

ID マッピングテーブルは、コラボレーションでマルチパーティ ID マッピング AWS Clean Rooms を有効にする のリソースです。

ID マッピングテーブルを作成する前に、まずソースデータとターゲットデータの両方を ID 名前空間として設定する必要があります。

ID マッピングテーブルを作成したら、ID マッピングワークフローを使用してソース ID 名前空間をターゲット ID 名前空間に変換します。これを行うには、ルールベースの方法、またはプロバイダーサービスのトランスコーディング方法を使用します。

ID マッピングワークフローは、指定された ID マッピングワークフロー方法に基づいて、入力データソースから入力データターゲットにデータをマッピングするデータ処理ジョブです。このワークフローでは、ID マッピングテーブルが入力されます。

Note

ID マッピングテーブルは、Amazon S3 に保存され、AWS Glue テーブルにクロールされたデータセットからのみ作成できます。

ID マッピングワークフローには、ルールベースの ID マッピングまたはプロバイダーサービス ID マッピングの 2 つの方法があります。

- ルールベースの ID マッピング – 一致するルールを使用して、ソースからターゲットにファーストパーティータータを変換します。
- プロバイダーサービス ID マッピング – LiveRamp プロバイダーサービスを使用して、ソースからターゲットにサードパーティータータを変換します。

Note

現在サポートされているトランスコーディングサービスプロバイダーは LiveRamp です。を通じて LiveRamp のサブスクリプションを持つコラボレーションのメンバー AWS Data Exchange は、ID マッピングテーブルを作成できます。LiveRamp へのサブスクリプションをすでに持っているが、経由では持っていない場合は AWS Data Exchange、LiveRamp に連絡してプライベートオファーを取得してください。詳細については、「AWS Entity Resolution ユーザーガイド」の「[AWS Data Exchangeでプロバイダーサービスをサブスクライブする](#)」を参照してください。

トピック

- [新しい ID マッピングテーブルの作成とデータ入力](#)
- [既存の ID マッピングテーブルへのデータ入力](#)
- [ID マッピングテーブルの編集](#)
- [ID マッピングテーブルの削除](#)

新しい ID マッピングテーブルの作成とデータ入力

前提条件

ID マッピングテーブルを作成する前に、以下を確認してください。

- 関連付けられた ID 名前空間のソースとターゲット
- ルールベースの ID マッピングまたはプロバイダーサービス ID マッピング用に設定された ID 名前空間

次の 2 種類の ID マッピングテーブルを作成できます。

- ルールベース – 一致するルールを使用してファーストパーティデータを翻訳します
- プロバイダーサービス – LiveRamp を使用して RampIDs

ID マッピングテーブルを作成したら、ID マッピングワークフローを実行してすぐに入力するか、後でテーブルの入力を待つことができます。

ID マッピングテーブルに正常にデータが入力されたら、ID マッピングテーブルでマルチテーブル結合クエリを実行して、sourceId を targetId に結合し、データを分析できます。

トピック

- [ID マッピングテーブルを作成する \(ルールベース\)](#)
- [ID マッピングテーブルを作成する \(プロバイダーサービス\)](#)

ID マッピングテーブルを作成する (ルールベース)

このトピックでは、一致するルールを使用してファーストパーティデータをソースからターゲットに変換する ID マッピングテーブルを作成するプロセスについて説明します。

ルールベースの ID マッピングテーブルを作成する場合、増分処理を有効にすることで、ワークフロー内の新規、更新、または削除されたレコードのみを処理できます。

ルールベースのメソッドを使用して新しい ID マッピングテーブルを作成してデータを入力するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. ナビゲーションペインで、コラボレーションを選択します。
3. コラボレーションを選択し、エンティティ解決タブを選択します。
4. [ID マッピングテーブルを作成] を選択します。
5. ID マッピング設定で、次のいずれかを実行します。
 - 新しいワークフローを作成するには、新しい ID マッピングワークフローの作成を選択したままにします。
 - 既存のワークフローを使用するには、チェックボックスをオフにしてドロップダウンリストからワークフローを選択し、ステップ 9 に進みます。
6. ID データで、ソースとターゲットを表示または設定します。
 - 単一の ID 名前空間ペアの場合: 事前に選択されたソースとターゲットを確認します。

- 複数の ID 名前空間の場合: ドロップダウンリストからソースとターゲットを選択します。
7. (オプション) 増分処理をオンにして、ワークフロー内の新規、更新、または削除されたレコードのみを処理します。

AWS Entity Resolution は、ID マッピングテーブル全体を再作成するのではなく、ソース ID またはターゲット ID 名前空間の新規、更新、または削除されたレコードのみを処理します。

このオプションを選択したままにすると、は ID マッピングテーブルでデフォルトのバッチ処理 ID マッピングワークフロー AWS Entity Resolution を実行します。

8. ルールパラメータで、以下を設定します。

- ルールコントロール – ターゲットまたはソースが一致するルールを提供するかどうかを選択します。

[ルールを表示] をオンにすると、ルールを表示できます。

ルールコントロールは、ID マッピングワークフローで使用されるソース ID とターゲット ID 名前空間の間で互換性がある必要があります。例えば、ソース ID 名前空間がルールをターゲットに制限するが、ターゲット ID 名前空間がルールをソースに制限する場合、エラーが発生します。

- [比較タイプ] は、自動的に [複数の入力フィールド] に設定されます。

これは、両方の参加者が以前にこのオプションを選択したためです。

- レコードマッチング
 - 1 つのソースから 1 つのターゲット – ターゲットごとに 1 つの一致するレコードを保存します
 - 1 つのターゲットへの多くのソース – ターゲットごとに一致するすべてのレコードを保存します

 Note

ソース ID 名前空間とターゲット ID 名前空間に指定されている制限は、互換性がある必要があります。

9. ID マッピングの詳細については、以下を設定します。

- a. ID マッピングテーブル名を入力するか、デフォルト名のままにします。
- b. (オプション) ID マッピングテーブルの [説明] を入力します。

この説明はクエリの作成に役立ちます。

10. AWS Clean Rooms アクセスするには、次のいずれかを選択します。

- アクセス許可ポリシーの追加と管理 AWS Clean Rooms を許可する – サービスロールを自動的に作成します。
- アクセス許可を手動で追加および管理 – リソースポリシーを確認して変更するか、ポリシーステートメントの追加を選択します。

 Note

ロールポリシーを変更できない場合は、がサービスロールのポリシーを見つけ AWS Clean Rooms られなかったことを示すエラーメッセージが表示されます。

11. AWS Entity Resolution アクセスするには、次のいずれかを選択します。

このセクションは、新しい ID マッピングテーブルを作成する場合にのみ表示されます。

- 新しいサービスロールを作成して使用
 - デフォルトの [サービスロール名] は `entityresolution-id-mapping-workflow-
<timestamp>` です。
 - (オプション) 暗号化されたデータの場合は、このデータは KMS キーによって暗号化され、AWS KMS キーを入力します。
- 既存のサービスロールを使用
 - ドロップダウンリストから既存のサービスロール名を選択するか、ロール ARN を入力します。

ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。

[IAM で表示] 外部リンクを選択してサービスロールを表示します。

既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。

デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

- (オプション) 必要なアクセス許可を持つ事前設定済みポリシーをこのロールに追加して、必要なアクセス許可をロールにアタッチします。

ルールを変更したりポリシーを作成したりするには、アクセス許可が必要です。

12. (オプション) 追加設定で、以下を設定します。

a. ID マッピングテーブルの設定

- カスタム暗号化を有効にするには、暗号化設定をカスタマイズを選択し、AWS KMS キーを入力します。

Note

この KMS キーは、内で KMS キーポリシー `cleanrooms.amazonaws.com` を使用するために必要なアクセス許可 `AWS Entity Resolution` をに付与する必要があります。ID マッピングワークフローで暗号化を使用するために必要なアクセス許可の詳細については、「AWS Entity Resolution ユーザーガイド」の「[AWS Entity Resolution のワークフロージョブロールを作成する](#)」を参照してください。

- タグを追加するには、新しいタグを追加を選択し、キーと値のペアを入力します

b. ID マッピングワークフロー設定 (新しいワークフローのみ):

- 異なる名前を使用するには、「同じ ID マッピングテーブルの名前と説明を保持」と「新しい値」を入力します。
- タグを追加するには、新しいタグを追加を選択し、キーと値のペアを入力します

13. 次のいずれかを選択します。

- ID マッピングテーブルの作成 – 後で入力できる空のテーブルを作成します ([既存の ID マッピングテーブルへのデータ入力](#))
- ID マッピングテーブルの作成と入力 – テーブルを作成してすぐに入力します (数時間かかる場合があります)。

ID マッピングワークフロープロセスが開始されます。このプロセス中、ID マッピングテーブルには変換された ID が入力されます。ID マッピングワークフローの処理には数時間かかる場合があります。

ID マッピングテーブルに正常にデータが入力されたら、[ID マッピングテーブルのクエリを実行](#)して `sourceId` を `targetId` と結合し、データを分析できます。

ID マッピングテーブルを作成する (プロバイダーサービス)

このトピックでは、プロバイダーサービス (LiveRamp) を使用する ID マッピングテーブルを作成するプロセスについて説明します。LiveRamp プロバイダーサービスは、保持された RampID または派生された RampID を使用して、ソース RampID のセットを変換します。

プロバイダーサービスのメソッドを使用して新しい ID マッピングテーブルを作成するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. ナビゲーションペインで、コラボレーションを選択します。
3. コラボレーションを選択し、エンティティ解決タブを選択します。
4. [ID マッピングテーブルを作成] を選択します。
5. ID マッピング設定で、次のいずれかを実行します。
 - 新しいワークフローを作成するには、新しい ID マッピングワークフローの作成を選択したままにします。
 - 既存のワークフローを使用するには、チェックボックスをオフにしてドロップダウンリストからワークフローを選択し、ステップ 9 に進みます。
6. ID データで、ソースとターゲットを表示または設定します。
 - 単一の ID 名前空間ペアの場合: 事前に選択したソースとターゲットを確認します。
 - 複数の ID 名前空間の場合: ドロップダウンリストからソースとターゲットを選択します。
7. [メソッド]で、選択した ID マッピングワークフローメソッドが [LiveRamp トランスコーディング] であることを確認します。
8. LiveRamp 設定の場合は、次のいずれかを実行します。
 - LiveRamp ID マネージャー ARN と LiveRamp シークレットマネージャー ARN を入力します。
 - 既存のワークフローからインポートを選択します。
9. ID マッピングの詳細については、以下を設定します。
 - a. ID マッピングテーブル名を入力するか、デフォルト名のままにします。
 - b. (オプション) ID マッピングテーブルの [説明] を入力します。

この説明はクエリの作成に役立ちます。

10. AWS Clean Rooms アクセスするには、次のいずれかを選択します。

- アクセス許可ポリシーの追加と管理 AWS Clean Rooms を許可する – サービスロールを自動的に作成します。
- アクセス許可を手動で追加および管理 – リソースポリシーを確認して変更するか、ポリシーステートメントの追加を選択します。

 Note

ロールポリシーを変更できない場合は、`ガ`サービスロールのポリシーを見つけ AWS Clean Rooms られなかったことを示すエラーメッセージが表示されます。

11. AWS Entity Resolution アクセスの場合は、1 つ選択します。

このセクションは、新しい ID マッピングテーブルを作成する場合にのみ表示されます。

- 新しいサービスロールを作成して使用
 - デフォルトの [サービスロール名] は `entityresolution-id-mapping-workflow-<timestamp>` です。
 - (オプション) 暗号化されたデータの場合は、このデータは KMS キーによって暗号化され、AWS KMS キーを入力します。
- 既存のサービスロールを使用
 - ドロップダウンリストから既存のサービスロール名を選択するか、ロール ARN を入力します。

ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。

[IAM で表示] 外部リンクを選択してサービスロールを表示します。

既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。

デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

- (オプション) 必要なアクセス許可を持つ事前設定済みポリシーをこのロールに追加して、必要なアクセス許可をロールにアタッチします。

ロールを変更したりポリシーを作成したりするには、アクセス許可が必要です。

12. (オプション) 追加設定で以下を設定します。

a. ID マッピングテーブルの設定

- カスタム暗号化を有効にするには、暗号化設定をカスタマイズを選択し、AWS KMS キーを入力します。

Note

この KMS キーは、内で KMS キーポリシー `cleanrooms.amazonaws.com` を使用するために必要なアクセス許可 `AWS Entity Resolution` をに付与する必要があります。ID マッピングワークフローで暗号化を使用するために必要なアクセス許可の詳細については、「AWS Entity Resolution ユーザーガイド」の「[AWS Entity Resolution のワークフロージョブロールを作成する](#)」を参照してください。

- タグを追加するには、新しいタグを追加を選択し、キーと値のペアを入力します。

b. ID マッピングワークフロー設定 (新しいワークフローのみ):

- 別の名前を使用するには、「同じ ID マッピングテーブルの名前と説明を保持」と「新しい値」を入力します。
- タグを追加するには、新しいタグを追加を選択し、キーと値のペアを入力します。

13. 次のいずれかを選択します。

- ID マッピングテーブルの作成 – 後で入力できる空のテーブルを作成します ([既存の ID マッピングテーブルへのデータ入力](#))
- ID マッピングテーブルの作成と入力 – テーブルを作成してすぐに入力します (数時間かかる場合があります)。

ID マッピングワークフロープロセスが開始されます。このプロセス中、ID マッピングテーブルには変換された ID が入力されます。ID マッピングワークフローの処理には数時間かかる場合があります。

ID マッピングテーブルに正常にデータが入力されたら、[ID マッピングテーブルのクエリを実行](#)して `sourceId` を `targetId` と結合し、データを分析できます。

既存の ID マッピングテーブルへのデータ入力

ID 名前空間に新しいデータが追加されたら、このワークフローを使用します。[ID マッピングテーブルの作成](#)時に増分処理を有効にすることを選択した場合、ID マッピングテーブル全体を再作成するのではなく、ソース ID 名前空間またはターゲット ID 名前空間のいずれかで新規、更新、または削除されたレコードのみを処理できます。

既存の ID マッピングテーブルにデータを入力するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms/> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. エンティティ解決タブに移動します。
5. ID マッピングテーブルセクションで、ID マッピングテーブルを選択します。
6. この ID マッピングテーブルの増分処理を有効にしていない場合は、ID マッピングテーブルを選択し、入力を選択します。
7. この ID マッピングテーブルの増分処理を有効にしている場合は、 で入力を選択し、次のいずれかを選択します。

- 増分処理 – ソース ID またはターゲット ID 名前空間のいずれかで、新規、更新、または削除されたレコードのみを処理します。

頻繁な更新、日次実行、またはリアルタイムのデータ同期に推奨されます。

- バッチ処理 – ID マッピングテーブル全体を処理します。

初期設定、定期的なフル更新、またはソース ID 名前空間とターゲット ID 名前空間の両方で大幅な変更が発生した場合に推奨されます。

- 削除のみの処理 – ソース ID 名前空間から削除されたレコードのみを処理し、それに応じてターゲット ID 名前空間を更新します。

削除をすばやく同期するために推奨されます。

8. ID マッピングワークフロープロセスが開始されます。

このプロセス中、ID マッピングテーブルにはトランスコーディングされた ID が入力されます。ID マッピングワークフローの処理には数時間かかる場合があります。

ID マッピングテーブルに正常にデータが入力されたら、[ID マッピングテーブルのクエリを実行](#)して sourceId を targetId と結合できます。

ID マッピングテーブルの編集

コラボレーションメンバーは、作成した ID マッピングテーブルを編集できます。

ID マッピングテーブルを編集するには

1. にサインイン AWS マネジメントコンソール し、で[AWS Clean Rooms コンソール](#)を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [エンティティ解決] タブを選択します。
5. [ID マッピングテーブル] で、テーブルを選択します。
6. ID マッピングテーブルの詳細ページで、下へスクロールして [ID マッピングテーブルの詳細] を表示します。
7. [編集] を選択します。
8. [ID マッピングテーブルを編集] ページで、[説明] または [サービスアクセス情報] を更新します。
9. [Save changes] (変更の保存) をクリックします。

ID マッピングテーブルの削除

コラボレーションメンバーは、作成した ID マッピングテーブルを削除できます。このアクションにより、クエリを行えるメンバーはテーブルでクエリを実行できなくなります。

Warning

マッピングテーブルを削除すると、入力されたデータは完全に削除されます。

ID マッピングテーブルを削除するには

1. にサインイン AWS マネジメントコンソール し、で[AWS Clean Rooms コンソール](#)を開きます AWS アカウント (まだ開いていない場合)。

2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [エンティティ解決] タブを選択します。
5. [ID マッピングテーブル] で、テーブルを選択します。
6. ID マッピングテーブルの詳細ページで、下へスクロールして [ID マッピングテーブル] を表示します。
7. ID マッピングテーブルを選択し、[削除] を選択します。
8. ID マッピングテーブルを削除してよい場合は、[削除] を選択します。

の分析テンプレート AWS Clean Rooms

分析テンプレートは [のカスタム分析ルール AWS Clean Rooms](#) で使用します。分析テンプレートを使用すると、同じクエリを再利用するのに役立つパラメータを定義できます。は、リテラル値を持つパラメータ化のサブセット AWS Clean Rooms をサポートします。

分析テンプレートはコラボレーションごとに固有となります。各コラボレーションのメンバーは、そのコラボレーション内のクエリのみを表示できます。コラボレーションで差分プライバシーを使用する予定がある場合は、分析テンプレートが AWS Clean Rooms 差分プライバシーの [汎用クエリ構造](#) と互換性があることを確認する必要があります。

分析テンプレートは、SQL コードを使用するか、Spark の Python コードを使用するかの 2 つの方法で作成できます。

トピック

- [SQL 分析テンプレート](#)
- [PySpark 分析テンプレート](#)
- [PySpark 分析テンプレートのトラブルシューティング](#)

SQL 分析テンプレート

SQL 分析テンプレートを使用すると、コラボレーション内の異なるデータセット間でデータをクエリおよび分析できます。これらのテンプレートを使用して、対象者の重複の特定や集計メトリクスの計算など、さまざまなタイプの分析を実行できます。

SQL 分析テンプレートを使用すると、次のことができます。

- 標準 SQL クエリの書き込み
- パラメータを追加してクエリを動的にする
- 特定の列とテーブルへのアクセスを制御する
- 機密データの集約要件を設定する
- カスタム機械学習 (ML) モデル用のプライバシー強化合成データセットを生成するための入力データを定義する

トピック

- [SQL 分析テンプレートの作成](#)

- [SQL 分析テンプレートの確認](#)

SQL 分析テンプレートの作成

前提条件

SQL 分析テンプレートを作成する前に、以下が必要です。

- アクティブな AWS Clean Rooms コラボレーション
- コラボレーションで少なくとも 1 つの設定済みテーブルへのアクセス

でテーブルを設定する方法については AWS Clean Rooms、「」を参照してください [で設定済みテーブルを作成する AWS Clean Rooms](#)。

- 分析テンプレートを作成するアクセス許可
- SQL クエリ構文の基本知識

次の手順では、[AWS Clean Rooms コンソール](#)を使用して SQL 分析テンプレートを作成するプロセスについて説明します。

AWS SDKs を使用して SQL 分析テンプレートを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

SQL 分析テンプレートを作成するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエイターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テンプレート] タブで、[自分で作成した分析テンプレート] セクションに移動します。
5. [分析テンプレートを作成] を選択します。
6. 分析テンプレートの作成ページの「詳細」で、
 - a. 分析テンプレートの名前を入力します。
 - b. (オプション) [説明] を入力します。
 - c. Format では、SQL オプションを選択したままにします。
7. [テーブル] には、コラボレーションに関連する設定済みテーブルが表示されます。

8. [定義] で以下の操作を行います。

- a. 分析テンプレートの定義を入力します。
- b. [インポート元] を選択して定義をインポートします。
- c. (オプション) SQL エディタで、パラメータ名の前にコロン (:) を入力してパラメータを指定します。

例えば、次のようになります。

```
WHERE table1.date + :date_period > table1.date
```

9. 以前にパラメータを追加したことがある場合は、[パラメータ – オプション] で、パラメータ名ごとに [タイプ] と [既定値] (オプション) を選択します。
10. 合成データの場合、モデルトレーニング用の合成データを生成する場合は、分析テンプレートの出力を合成にするチェックボックスをオンにします。

詳細については、[「プライバシーが強化された合成データセットの生成」](#)を参照してください。

- a. 列分類で、ドロップダウンリストから列を選択します。少なくとも 5 つの列が必要です。
 - i. ドロップダウンリストから分類を選択します。これにより、各列のデータ型が識別されます。

分類タイプには以下が含まれます。

 - 数値 – 測定値やカウントなどの連続した数値
 - カテゴリ – ラベルやタイプなどの個別の値やカテゴリ
 - ii. 列を削除するには、削除を選択します。
 - iii. 別の列を追加するには、別の列の追加を選択します。ドロップダウンリストから列と分類を選択します。
 - iv. 予測値で、ドロップダウンリストから列を選択します。これは、合成データセットでトレーニングされた後、カスタムモデルが予測に使用する列です。
- b. 詳細設定では、プライバシーレベルとプライバシーしきい値を設定できます。ニーズに合わせて設定を調整します。
 - プライバシーレベルでは、イプシロン値を入力して、生成されたデータセットのプライバシーを保護するために合成モデルが追加するノイズの量を決定します。値は 0.0001 ~ 10 である必要があります。

- 値を小さくするとノイズが増え、プライバシー保護が強化されますが、このデータでトレーニングされたダウンストリームカスタムモデルのユーティリティが低下する可能性があります。
- 値を大きくするとノイズが少なくなり、精度が向上しますが、プライバシー保護が低下する可能性があります。

プライバシーしきい値には、メンバーシップ推論攻撃が元のデータセットのメンバーを識別できる最大許容確率を入力します。値は 50.0 ~ 100 の間でなければなりません。

- スコアが 50% の場合、メンバーシップ推論攻撃では、ランダムな推測よりもメンバーと非メンバーを正常に区別できないことを示します。
- プライバシー制限がない場合は、100% と入力します。

最適な値は、特定のユースケースとプライバシー要件によって異なります。プライバシーしきい値を超えると、ML 入力チャネルの作成は失敗し、合成データセットを使用してモデルをトレーニングすることはできません。

 Warning

合成データ生成は、特定の個人が元のデータセットに存在するか、それらの個人の学習属性が存在するかにかかわらず、個々の属性の推測から保護します。ただし、個人を特定できる情報 (PII) を含む元のデータセットのリテラル値が合成データセットに表示されるのを防ぐことはできません。

1 つのデータサブジェクトのみに関連付けられている入力データセット内の値は、データセットを再識別する可能性があるため、避けることをお勧めします。たとえば、郵便番号にユーザーが 1 人しかいない場合、合成データセットにその郵便番号が存在すると、そのユーザーが元のデータセットに属していたことが確認されます。高精度値の切り捨てや、まれなカタログを他のカタログに置き換えるなどの手法を使用して、このリスクを軽減できます。これらの変換は、ML 入力チャネルの作成に使用されるクエリの一部にすることができます。

11. リソースのタグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
12. [作成] を選択します。

13. これで、[分析テンプレートを確認できることを](#)コラボレーションメンバーに通知する準備ができました。(自身のデータにクエリを実行する場合は省略可能)

SQL 分析テンプレートの確認

コラボレーションメンバーが SQLanalysisテンプレートを作成したら、レビューして承認できます。分析テンプレートが承認されたら、 のクエリで使用できます AWS Clean Rooms。

Note

分析コードをコラボレーションに取り込むときは、次の点に注意してください。

- AWS Clean Rooms は、分析コードの動作を検証または保証しません。
 - 特定の動作を確認する必要がある場合は、コラボレーションパートナーのコードを直接確認するか、信頼できるサードパーティーの監査人と協力して確認してください。
- 共有セキュリティモデルの場合:
 - ユーザー (顧客) は、環境で実行されているコードのセキュリティに責任を負います。
 - AWS Clean Rooms は環境のセキュリティを担当し、
 - 承認されたコード実行のみ
 - 指定された設定済みテーブルのみにアクセス可能
 - 唯一の出力先は、結果レシーバーの S3 バケットです。

AWS Clean Rooms コンソールを使用して SQL 分析テンプレートを確認するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエイターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テンプレート] タブで、[他のメンバーが作成した分析テンプレート] セクションに移動します。
5. [実行可能ステータス] が [いいえ - ユーザーのレビューが必要] の分析テンプレートを選択します。
6. [Review] (レビュー) を選択します。
7. 分析ルールの [概要]、[定義]、および [パラメータ] (ある場合) を確認します。

8. [定義で参照されるテーブル] に表示されている設定済みのテーブルを確認します。

各テーブルの横の [ステータス] には、[テンプレートは許可されていません] と表示されています。

9. テーブルを選択します。

オプション:	選択内容
分析テンプレートを承認する	[テーブルでテンプレートを許可] を選択し、[許可] を選択して承認を確定します。
分析テンプレートを承認しない	[許可しない] を選択します。

これで、SQL 分析テンプレートを使用して設定済みテーブルをクエリする準備ができました。詳細については、「[SQL クエリの実行](#)」を参照してください。

PySpark 分析テンプレート

PySpark 分析テンプレートでは、カスタムライブラリとオープンソースライブラリを使用するには、Python ユーザースクリプトとオプションの仮想環境が必要です。これらのファイルはアーティファクトと呼ばれます。

分析テンプレートを作成する前に、まずアーティファクトを作成し、そのアーティファクトを Amazon S3 バケットに保存します。は、分析ジョブの実行時にこれらのアーティファクト AWS Clean Rooms を使用します。は、ジョブの実行時に AWS Clean Rooms のみアーティファクトにアクセスします。

PySpark 分析テンプレートでコードを実行する前に、はアーティファクトを次のように AWS Clean Rooms 検証します。

- テンプレートの作成時に使用する特定の S3 オブジェクトバージョンの確認
- アーティファクトの SHA-256 ハッシュの検証
- アーティファクトが変更または削除されたジョブの失敗

Note

の特定の PySpark 分析テンプレートのすべての結合アーティファクトの最大サイズ AWS Clean Rooms は 1 GB です。

PySpark 分析テンプレートのセキュリティ

安全なコンピューティング環境を維持するために、は 2 層コンピューティングアーキテクチャ AWS Clean Rooms を使用して、ユーザーコードをシステムオペレーションから分離します。このアーキテクチャは、Membrane と呼ばれる Amazon EMR Serverless Fine Grained Access Control テクノロジーに基づいています。詳細については、[「Membrane — Safe and performant data access controls in Apache Spark in the presence of imperative code」](#)を参照してください。

コンピューティング環境コンポーネントは、個別のユーザースペースとシステムスペースに分割されます。ユーザースペースは、PySpark 分析テンプレートで PySpark コードを実行します。は、システムスペース AWS Clean Rooms を使用してジョブの実行を有効にします。これには、顧客が提供するサービスロールを使用してデータを読み取ってジョブを実行し、列の許可リストを実装することが含まれます。このアーキテクチャの結果として、少数の Spark SQL および PySpark DataFrames APIs を含むシステムスペースに影響を与える顧客の PySpark コードがブロックされます。

での PySpark の制限 AWS Clean Rooms

顧客が承認済みの PySpark 分析テンプレートを送信すると、は顧客がアクセスできない独自の安全なコンピューティング環境でそれ AWS Clean Rooms を実行します。コンピューティング環境は、安全なコンピューティング環境を維持するために、ユーザースペースとシステムスペースを備えたコンピューティングアーキテクチャを実装します。詳細については、[「PySpark 分析テンプレートのセキュリティ」](#)を参照してください。

PySpark を使用する前に、次の制限事項を考慮してください AWS Clean Rooms。

制限事項

- DataFrame 出力のみがサポートされています
- ジョブ実行ごとに 1 つの Spark セッション

サポートされていない機能

- データ管理

- Iceberg テーブル形式
- LakeFormation マネージドテーブル
- 耐障害性のある分散データセット (RDD)
- Spark ストリーミング
- ネストされた列のアクセスコントロール
- カスタム関数と拡張機能
 - ユーザー定義テーブル関数 (UDTFs)
 - HiveUDFs
 - ユーザー定義関数のカスタムクラス
 - カスタムデータソース
 - 追加の JAR ファイル:
 - Spark 拡張機能
 - Connector
 - メタストア設定
- モニタリングと分析
 - Spark ログ記録
 - Spark UI
 - ANALYZE TABLE コマンド

Important

これらの制限は、ユーザースペースとシステムスペース間のセキュリティ分離を維持するために設定されています。

コラボレーション設定に関係なく、すべての制限が適用されます。

今後の更新により、セキュリティ評価に基づく追加機能のサポートが追加される場合があります。

ベストプラクティス

PySpark 分析テンプレートを作成するときは、次のベストプラクティスをお勧めします。

- 分析テンプレートは、[での PySpark の制限 AWS Clean Rooms](#) を念頭に置いて設計します。

- まず、開発環境でコードをテストします。
- サポートされている DataFrame オペレーションのみを使用します。
- DataFrame の制限に対応するために出力構造を計画します。

アーティファクトを管理するには、次のベストプラクティスをお勧めします。

- すべての PySpark 分析テンプレートアーティファクトを専用の S3 バケットまたはプレフィックスに保持します。
- さまざまなアーティファクトバージョンにクリアバージョンの命名を使用します。
- アーティファクトの更新が必要な場合は、新しい分析テンプレートを作成します。
- どのテンプレートがどのアーティファクトバージョンを使用するかのインベントリを維持します。

Spark コードを記述する方法の詳細については、以下を参照してください。

- [Apache Spark の例](#)
- Amazon EMR リリースガイドの「[Spark アプリケーションの記述](#)」
- [チュートリアル: ユーザーガイドの AWS Glue for Spark スクリプトの記述](#) AWS Glue

以下のトピックでは、分析テンプレートを作成して確認する前に Python ユーザースクリプトとライブラリを作成する方法について説明します。

トピック

- [ユーザースクリプトの作成](#)
- [PySpark 分析テンプレートでのパラメータの操作](#)
- [仮想環境の作成 \(オプション\)](#)
- [ユーザースクリプトと仮想環境を S3 に保存する](#)
- [PySpark 分析テンプレートの作成](#)
- [PySpark 分析テンプレートの確認](#)

ユーザースクリプトの作成

ユーザースクリプトには、エントリポイント関数 (つまり、ハンドラー) が含まれている必要があります。ユーザースクリプトファイルに任意の有効な Python ファイル名を付けることができます。

次の手順では、ユーザースクリプトを作成して PySpark 分析のコア機能を定義する方法について説明します。

前提条件

- PySpark 1.0 (Python 3.11 および Spark 3.5.3 に対応)
- Amazon S3 のデータセットは、定義した Spark セッションで設定されたテーブルの関連付けとしてのみ読み取ることができます。
- コードが Amazon S3 とを直接呼び出すことはできません AWS Glue
- コードがネットワークコールを実行できない

ユーザースクリプトを作成するには

1. 選択したテキストエディタまたは統合開発環境 (IDE) を開きます。

Python ファイルをサポートする任意のテキストエディタまたは IDE (Visual Studio Code、PyCharm、Notepad++ など) を使用できます。

2. 選択した名前で新しい Python ファイルを作成します (例: **my_analysis.py**)。
3. コンテキストオブジェクトパラメータを受け入れるエントリポイント関数を定義します。

```
def entrypoint(context)
```

context オブジェクトパラメータは、基本的な Spark コンポーネント、参照テーブル、および分析パラメータへのアクセスを提供するディクショナリです。以下を含みます。

を介した Spark セッションアクセス context['sparkSession']

を介した参照テーブル context['referencedTables']

による分析パラメータ context['analysisParameters'] (テンプレートでパラメータが定義されている場合)

4. エントリポイント関数の結果を定義します。

```
return results
```

は、ファイル名の結果ディクショナリを含むオブジェクトを出力 DataFrame に返す results 必要があります。

Note

AWS Clean Rooms は、DataFrame オブジェクトを結果レシーバーの S3 バケットに自動的に書き込みます。

5. これで次の作業に進むことができます。

- a. このユーザースクリプトを S3 に保存します。詳細については、「[ユーザースクリプトと仮想環境を S3 に保存する](#)」を参照してください。
- b. ユーザースクリプトに必要な追加のライブラリをサポートするオプションの仮想環境を作成します。詳細については、「[仮想環境の作成 \(オプション\)](#)」を参照してください。

Example例 1

<caption>The following example demonstrates a generic user script for a PySpark analysis template.</caption>

```
# File name: my_analysis.py

def entrypoint(context):
    try:
        # Access Spark session
        spark = context['sparkSession']

        # Access input tables
        input_table1 = context['referencedTables']['table1_name']
        input_table2 = context['referencedTables']['table2_name']

        # Example data processing operations
        output_df1 = input_table1.select("column1", "column2")
        output_df2 = input_table2.join(input_table1, "join_key")
        output_df3 = input_table1.groupBy("category").count()

        # Return results - each key creates a separate output folder
        return {
            "results": {
                "output1": output_df1,          # Creates output1/ folder
                "output2": output_df2,          # Creates output2/ folder
                "analysis_summary": output_df3 # Creates analysis_summary/ folder
            }
        }
    }
```

```
except Exception as e:
    print(f"Error in main function: {str(e)}")
    raise e
```

この例のフォルダ構造は次のとおりです。

```
analysis_results/
#
### output1/ # Basic selected columns
# ### part-00000.parquet
# ### _SUCCESS
#
### output2/ # Joined data
# ### part-00000.parquet
# ### _SUCCESS
#
### analysis_summary/ # Aggregated results
### part-00000.parquet
### _SUCCESS
```

Example例 2

<caption>The following example demonstrates a more complex user script for a PySpark analysis template.</caption>

```
def entrypoint(context):
    try:
        # Get DataFrames from context
        emp_df = context['referencedTables']['employees']
        dept_df = context['referencedTables']['departments']

        # Apply Transformations
        emp_dept_df = emp_df.join(
            dept_df,
            on="dept_id",
            how="left"
        ).select(
            "emp_id",
            "name",
            "salary",
            "dept_name"
        )
```

```
# Return Dataframes
return {
    "results": {
        "outputTable": emp_dept_df
    }
}

except Exception as e:
    print(f"Error in entrypoint function: {str(e)}")
    raise e
```

PySpark 分析テンプレートでのパラメータの操作

パラメータは、ジョブの送信時に異なる値を指定できるようにすることで、PySpark 分析テンプレートの柔軟性を高めます。パラメータは、エントリポイント関数に渡されるコンテキストオブジェクトからアクセスできます。

Note

パラメータは、任意のコンテンツを含めることができるユーザー提供の文字列です。

- コードを確認して、分析で予期しない動作が発生しないように、パラメータが安全に処理されていることを確認します。
- 送信時に提供されるパラメータ値に関係なく、安全に機能するようにパラメータ処理を設計します。

パラメータへのアクセス

パラメータは `context['analysisParameters']` ディクショナリで使用できます。すべてのパラメータ値は文字列です。

Example パラメータへの安全なアクセス

```
def entrypoint(context):
    # Access parameters from context
    parameters = context['analysisParameters']
    threshold = parameters['threshold']
    table_name = parameters['table_name']
```

```
# Continue with analysis using parameters
spark = context['sparkSession']
input_df = context['referencedTables'][table_name]

# Convert threshold value
threshold_val = int(threshold)

# Use parameter in DataFrame operation
filtered_df = input_df.filter(input_df.amount > threshold_val)

return {
    "results": {
        "output": filtered_df
    }
}
```

パラメータのセキュリティベストプラクティス

Warning

パラメータは、任意のコンテンツを含めることができるユーザー提供の文字列です。分析コードのセキュリティの脆弱性を防ぐために、パラメータを安全に処理する必要があります。

以下を回避するための安全でないパラメータ処理パターン:

- パラメータをコードとして実行する – パラメータ値`exec()`で `eval()`または `eval()` を使用しないでください

```
# UNSAFE - Don't do this
eval(parameters['expression']) # Can execute arbitrary code
```

- SQL 文字列補間 – パラメータを SQL 文字列に直接連結しないでください

```
# UNSAFE - Don't do this
sql = f"SELECT * FROM table WHERE column = '{parameters['value']}'" # SQL injection risk
```

- 安全でないファイルパスオペレーション – 検証なしでファイルシステムオペレーションでパラメータを直接使用しない

```
# UNSAFE - Don't do this
file_path = f"/data/{parameters['filename']}" # Path traversal risk
```

安全なパラメータ処理パターン:

- DataFrame オペレーションでパラメータを使用する – Spark DataFrames はパラメータ値を安全に処理します

```
# SAFE - Use parameters in DataFrame operations
threshold = int(parameters['threshold'])
filtered_df = input_df.filter(input_df.value > threshold)
```

- パラメータ値を検証する – 使用前に、パラメータが予想される形式を満たしていることを確認します。

```
# SAFE - Validate parameters before use
def validate_date(date_str):
    try:
        from datetime import datetime
        datetime.strptime(date_str, '%Y-%m-%d')
        return True
    except ValueError:
        return False

date_param = parameters['date_filter'] or '2024-01-01'
if not validate_date(date_param):
    raise ValueError(f"Invalid date format: {date_param}")
```

- パラメータ値に許可リストを使用する – 可能であれば、既知の正常な値と照らし合わせてパラメータを検証します。

```
# SAFE - Use allowlists
allowed_columns = ['column1', 'column2', 'column3']
column_param = parameters['column_name']
if column_param not in allowed_columns:
    raise ValueError(f"Invalid column: {column_param}")
```

- エラー処理による型変換 — 文字列パラメータを予期される型に安全に変換する

```
# SAFE - Convert with error handling
```

```
try:
    batch_size = int(parameters['batch_size'] or '1000')
    if batch_size <= 0 or batch_size > 10000:
        raise ValueError(f"Batch size must be between 1 and 10000")
except ValueError as e:
    print(f"Invalid parameter: {e}")
    raise
```

Important

ジョブランナーが異なる値を提供する場合、パラメータはコードレビューをバイパスすることに注意してください。提供されるパラメータ値に関係なく、安全に機能するようにパラメータ処理を設計します。

完全なパラメータの例

Example PySpark スクリプトでのパラメータの安全な使用

```
def entrypoint(context):
    try:
        # Access Spark session and tables
        spark = context['sparkSession']
        input_table = context['referencedTables']['sales_data']

        # Access parameters - fail fast if analysisParameters missing
        parameters = context['analysisParameters']

        # Validate and convert numeric parameter (handles empty strings with default)
        try:
            threshold = int(parameters['threshold'] or '100')
            if threshold <= 0:
                raise ValueError("Threshold must be positive")
        except (ValueError, TypeError) as e:
            print(f"Invalid threshold parameter: {e}")
            raise

        # Validate date parameter (handles empty strings with default)
        date_filter = parameters['start_date'] or '2024-01-01'
        from datetime import datetime
        try:
```

```
        datetime.strptime(date_filter, '%Y-%m-%d')
    except ValueError:
        raise ValueError(f"Invalid date format: {date_filter}")

    # Use parameters safely in DataFrame operations
    filtered_df = input_table.filter(
        (input_table.amount > threshold) &
        (input_table.date >= date_filter)
    )

    result_df = filtered_df.groupBy("category").agg(
        {"amount": "sum"}
    )

    return {
        "results": {
            "filtered_results": result_df
        }
    }

except Exception as e:
    print(f"Error in analysis: {str(e)}")
    raise
```

仮想環境の作成 (オプション)

ユーザースクリプトに必要な追加のライブラリがある場合は、それらのライブラリを保存する仮想環境を作成するオプションがあります。追加のライブラリが必要ない場合は、このステップをスキップできます。

ネイティブ拡張機能を持つライブラリを使用する場合は、の PySpark が ARM64 アーキテクチャの Linux で AWS Clean Rooms 動作することに注意してください。

次の手順は、基本的な CLI コマンドを使用して仮想環境を作成する方法を示しています。

仮想環境を作成するには

1. ターミナルまたはコマンドプロンプトを開きます。
2. 次のコンテンツを追加します。

```
# create and activate a python virtual environment
python3 -m venv pyspark_venvsource
```

```
source pyspark_venvsource/bin/activate

# install the python packages
pip3 install pandas # add packages here

# package the virtual environment into an archive
pip3 install venv-pack
venv-pack -f -o pyspark_venv.tar.gz

# optionally, remove the virtual environment directory
deactivate
rm -fr pyspark_venvsource
```

3. これで、この仮想環境を S3 に保存する準備ができました。詳細については、「[ユーザースクリプトと仮想環境を S3 に保存する](#)」を参照してください。

Docker と Amazon ECR の操作の詳細については、「[Amazon ECR User ユーザーガイド](#)」を参照してください。

ユーザースクリプトと仮想環境を S3 に保存する

次の手順では、ユーザースクリプトとオプションの仮想環境を Amazon S3 に保存する方法を説明します。PySpark 分析テンプレートを作成する前に、このステップを完了します。

Important

分析テンプレートの作成後にアーティファクト (ユーザースクリプトまたは仮想環境) を変更または削除しないでください。

これを行うと、次のようになります。

- このテンプレートを使用する今後のすべての分析ジョブが失敗します。
- 新しいアーティファクトを使用して新しい分析テンプレートを作成する必要があります。
- 以前に完了した分析ジョブには影響しない

前提条件

- 適切なアクセス許可 AWS アカウント を持つ
- ユーザースクリプトファイル (など `my_analysis.py`)

- (オプション、存在する場合) 仮想環境パッケージ (.tar.gz ファイル)
- IAM ロールを作成または変更するためのアクセス

Console

コンソールを使用してユーザースクリプトと仮想環境を S3 に保存するには:

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。
2. 新しい S3 バケットを作成するか、既存のバケットを使用します。
3. バケットのバージョニングを有効にします。
 - a. バケットを選択します。
 - b. [プロパティ] を選択します。
 - c. バケットバージョニングセクションで、編集 を選択します。
 - d. 変更を有効にして保存します。
4. アーティファクトをアップロードし、SHA-256 ハッシュを有効にします。
 - a. バケットに移動します。
 - b. アップロード を選択します。
 - c. ファイルを追加 を選択し、ユーザースクリプトファイルを追加します。
 - d. (オプション、存在する場合) .tar.gz ファイルを追加します。
 - e. プロパティを展開します。
 - f. チェックサムで、チェックサム関数に SHA256 を選択します。
 - g. アップロード を選択します。
5. これで、PySpark 分析テンプレートを作成する準備ができました。

CLI

を使用してユーザースクリプトと仮想環境を S3 に保存するには AWS CLI:

1. 次のコマンドを実行します。

```
aws s3 cp --checksum-algorithm sha256 pyspark_venv.tar.gz s3://ARTIFACT-BUCKET/  
EXAMPLE-PREFIX/
```

2. これで、PySpark 分析テンプレートを作成する準備ができました。

Note

スクリプトまたは仮想環境を更新する必要がある場合:

1. 新しいバージョンを別のオブジェクトとしてアップロードします。
2. 新しいアーティファクトを使用して新しい分析テンプレートを作成します。
3. 古いテンプレートを廃止します。
4. 古いテンプレートがまだ必要な場合は、元のアーティファクトを S3 に保持します。

PySpark 分析テンプレートの作成

Note

パラメータは、任意のコンテンツを含めることができるユーザー提供の文字列です。

- コードを確認して、分析での予期しない動作を防ぐためにパラメータが安全に処理されていることを確認します。
- 送信時に提供されるパラメータ値に関係なく、安全に機能するようにパラメータ処理を設計します。

前提条件

PySpark 分析テンプレートを作成する前に、以下が必要です。

- アクティブな AWS Clean Rooms コラボレーションのメンバーシップ
- アクティブなコラボレーションで少なくとも 1 つの設定済みテーブルへのアクセス
- 分析テンプレートを作成するアクセス許可
- S3 で作成および保存される Python ユーザースクリプトと仮想環境
 - S3 バケットではバージョニングが有効になっています。詳細については、[S3 バケットでのバージョニングの使用](#)を参照してください。
 - S3 バケットは、アップロードされたアーティファクトの SHA-256 チェックサムを計算できません。詳細については、[「チェックサムの使用」](#)を参照してください。

- S3 バケットからコードを読み取るアクセス許可

必要なサービスロールの作成については、「」を参照してください[S3 バケットからコードを読み取るサービスロールを作成する \(PySpark 分析テンプレートロール\)](#)。

次の手順では、[AWS Clean Rooms コンソール](#)を使用して PySpark 分析テンプレートを作成するプロセスについて説明します。ユーザースクリプトと仮想環境ファイルを既に作成し、ユーザースクリプトと仮想環境ファイルを Amazon S3 バケットに保存していることを前提としています。

 Note

PySpark 分析テンプレートを作成するメンバーも、結果を受け取るメンバーである必要があります。

AWS SDKs を使用して PySpark 分析テンプレートを作成する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

PySpark 分析テンプレートを作成するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエイターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テンプレート] タブで、[自分で作成した分析テンプレート] セクションに移動します。
5. [分析テンプレートを作成] を選択します。
6. 分析テンプレートの作成ページの「詳細」で、
 - a. 分析テンプレートの名前を入力します。
 - b. (オプション) [説明] を入力します。
 - c. Format で、PySpark オプションを選択します。
7. [定義] で以下の操作を行います。
 - a. 続行する前に、前提条件を確認し、各前提条件が満たされていることを確認します。
 - b. エントリポイントファイルで、S3 バケットを入力するか、Browse S3 を選択します。
 - c. (オプション) ライブラリファイルで、S3 バケットを入力するか、S3 を参照を選択します。

8. パラメータ – 分析テンプレートを再利用可能にするためにパラメータを追加する場合はオプションです。

- a. [Add parameter] (パラメータの追加) を選択します。
- b. パラメータ名を入力します。

パラメータ名は文字またはアンダースコアで始まり、その後に英数字またはアンダースコアが続く必要があります。

- c. Type の場合、STRING は PySpark 分析テンプレートでサポートされている唯一のタイプとして自動的に選択されます。
- d. (オプション) パラメータのデフォルト値を入力します。

デフォルト値を指定すると、ジョブランナーはパラメータ値を明示的に指定せずにジョブを実行するときこの値を使用できます。

- e. さらにパラメータを追加するには、別のパラメータを追加を選択し、前のステップを繰り返します。

 Note

PySpark 分析テンプレートごとに最大 50 個のパラメータを定義できます。各パラメータ値は最大 1,000 文字です。

9. 定義で参照されるテーブルの場合、

- 定義で参照されているすべてのテーブルがコラボレーションに関連付けられている場合:
 - 定義で参照されているすべてのテーブルは、コラボレーションチェックボックスが選択されたままにします。
 - コラボレーションに関連付けられたテーブルで、定義で参照されているすべての関連テーブルを選択します。
- 定義で参照されているすべてのテーブルがコラボレーションに関連付けられていない場合:
 - 定義で参照されているすべてのテーブルがコラボレーションチェックボックスに関連付けられていることをクリアします。
 - コラボレーションに関連付けられたテーブルで、定義で参照されているすべての関連テーブルを選択します。
 - 後で関連付けるテーブルに、テーブル名を入力します。

- 別のテーブルを一覧表示する を選択して、別のテーブルを一覧表示します。

10. エラーメッセージ設定で、次のいずれかを選択します。

- 基本的なエラーメッセージ – 基になるデータを公開することなく、基本的なエラーメッセージを返します。本番ワークロードに推奨されます。
- 詳細なエラーメッセージ – 詳細なエラーメッセージを返し、トラブルシューティングを高速化します。開発およびテスト環境で推奨されます。個人を特定できる情報 (PII) などの機密データが公開される可能性があります。

 Note

詳細なエラーメッセージを使用する場合、すべてのデータプロバイダーメンバーがテンプレートに対してこの設定を承認する必要があります。

11. ドロップダウンリストから既存のサービスロール名を選択して、サービスアクセス許可を指定します。

1. ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。

ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。

2. [IAM で表示] 外部リンクを選択してサービスロールを表示します。

既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。

デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

 Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、 「」を参照してください [AWS の 管理ポリシー AWS Clean Rooms](#)。

- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに に対する十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。
- ロールポリシーを変更できない場合は、AWS Clean Rooms でサービスロールのポリシーが見つからなかったという内容のエラーメッセージが表示されます。

12. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
13. [作成] を選択します。
14. これで、コラボレーションメンバーに[分析テンプレートを確認](#)できることを通知する準備ができました。(自身のデータにクエリを実行する場合は省略可能)

Important

分析テンプレートの作成後にアーティファクト (ユーザースクリプトまたは仮想環境) を変更または削除しないでください。

これを行うと、次のようになります。

- このテンプレートを使用する今後のすべての分析ジョブが失敗します。
- 新しいアーティファクトを使用して新しい分析テンプレートを作成する必要があります。
- 以前に完了した分析ジョブには影響しません。

PySpark 分析テンプレートの確認

別のメンバーがコラボレーションで分析テンプレートを作成する場合は、使用する前に確認して承認する必要があります。

次の手順では、ルール、パラメータ、参照テーブルなど、PySpark 分析テンプレートを確認する方法を示します。コラボレーションメンバーは、テンプレートがデータ共有契約とセキュリティ要件に合致しているかどうかを評価します。

分析テンプレートが承認されたら、 のジョブで使用できます AWS Clean Rooms。

Note

分析コードをコラボレーションに取り込むときは、次の点に注意してください。

- AWS Clean Rooms は、分析コードの動作を検証または保証しません。
 - 特定の動作を確認する必要がある場合は、コラボレーションパートナーのコードを直接確認するか、信頼できるサードパーティーの監査人と協力して確認してください。
- AWS Clean Rooms は、PySpark 分析テンプレートにリストされているコードの SHA-256 ハッシュがPySpark 分析環境で実行されているコードと一致することを保証します。
- AWS Clean Rooms は、環境に持ち込む追加のライブラリの監査やセキュリティ分析を行いません。
- 共有セキュリティモデルの場合:
 - ユーザー (顧客) は、環境で実行されているコードのセキュリティに責任を負います。
 - ユーザー (顧客) は、環境に適したエラーメッセージ設定を行う責任があります。
 - AWS Clean Rooms は環境のセキュリティを担当し、
 - 承認されたコード実行のみ
 - 指定された設定済みテーブルのみにアクセスできます
 - 唯一の出力先は、結果レシーバーの S3 バケットです。

AWS Clean Rooms は、レビュー用にユーザースクリプトと仮想環境の SHA-256 ハッシュを生成します。ただし、実際のユーザースクリプトとライブラリには 内で直接アクセスできません AWS Clean Rooms。

共有されたユーザースクリプトとライブラリが分析テンプレートで参照されているものと同じであることを確認するには、共有されたファイルの SHA-256 ハッシュを作成し、 によって作成された分析テンプレートハッシュと比較します AWS Clean Rooms。コード実行のハッシュもジョブログに含まれます。

前提条件

- Linux/Unix オペレーティングシステムまたは Windows Subsystem for Linux (WSL)
- ハッシュするユーザースクリプトファイル
 - 分析テンプレート作成者が安全なチャネルを介してファイルを共有するようにリクエストします。
- によって作成された分析テンプレートハッシュ AWS Clean Rooms

AWS Clean Rooms コンソールを使用して PySpark 分析テンプレートを確認するには

1. にサインイン AWS マネジメントコンソール し、コラボレーションクリエイターとして機能する AWS アカウント で[AWS Clean Rooms コンソール](#)を開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. [テンプレート] タブで、[他のメンバーが作成した分析テンプレート] セクションに移動します。
5. [実行可能ステータス] が [いいえ - ユーザーのレビューが必要] の分析テンプレートを選択します。
6. [Review] (レビュー) を選択します。
7. 分析ルールの [概要]、[定義]、および [パラメータ] (ある場合) を確認します。

 Note

パラメータを使用すると、分析ランナーは送信時に異なる値を送信できます。分析テンプレートがパラメータをサポートしている場合は、コラボレーションパートナーのコードでパラメータ値がどのように使用されているかを確認し、要件を満たしていることを確認します。

8. 共有ユーザースクリプトとライブラリが分析テンプレートで参照されているものと同じであることを確認します。
 - a. 共有ファイルの SHA-256 ハッシュを作成し、 によって作成された分析テンプレートハッシュと比較します AWS Clean Rooms。

ハッシュを生成するには、ユーザースクリプトファイルを含むディレクトリに移動し、次のコマンドを実行します。

```
sha256sum your_script_filename.py
```

出力例:

```
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 my_analysis.py
```

- b. または、Amazon S3 チェックサム機能を使用することもできます。詳細については、[Amazon S3 ユーザーガイド](#)の「[Amazon S3 でのオブジェクトの整合性の確認](#)」を参照してください。 Amazon S3

- c. もう 1 つの方法は、実行されたコードのハッシュをジョブログに表示することです。
9. [定義で参照されるテーブル] に表示されている設定済みのテーブルを確認します。

各テーブルの横の [ステータス] には、[テンプレートは許可されていません] と表示されています。
10. テーブルを選択します。
 - a. 分析テンプレートを承認するには、テーブルでテンプレートを許可するを選択します。[許可] を選択して承認を確定します。
 - b. 承認を拒否するには、拒否を選択します。

分析テンプレートの承認を選択した場合、ジョブを実行できるメンバーはPySpark 分析テンプレートを使用して、設定されたテーブルで PySpark ジョブを実行できるようになりました。詳細については、「[PySpark ジョブの実行](#)」を参照してください。

PySpark 分析テンプレートのトラブルシューティング

PySpark 分析テンプレートを使用してジョブを実行すると、ジョブの初期化または実行中にエラーが発生する可能性があります。これらの障害は通常、スクリプト設定、データアクセス許可、または環境設定に関連しています。

PySpark の制限の詳細については、「」を参照してください [での PySpark の制限 AWS Clean Rooms](#)。

トピック

- [コードのトラブルシューティング](#)
- [分析テンプレートジョブが開始されない](#)
- [分析テンプレートジョブは開始されますが、処理中に失敗します](#)
- [仮想環境の設定が失敗する](#)

コードのトラブルシューティング

コードの開発とトラブルシューティングに役立つように、詳細なエラーメッセージを有効にし、AWS Clean Rooms 独自のテストデータを使用してジョブを実行する AWS アカウント ことで、独自のシミュレーションを行うことをお勧めします。

次の手順を使用して、Amazon EMR Serverless の AWS Clean Rooms で PySpark をシミュレートすることもできます。AWS Clean Rooms では PySpark と若干の違いがありますが、コードの実行方法がほとんどです。

EMR Serverless の で PySpark AWS Clean Rooms をシミュレートするには

1. Amazon S3 でデータセットを作成し、 でカタログ化し AWS Glue Data Catalog、Lake Formation アクセス許可を設定します。
2. カスタムロールを使用して S3 ロケーションを Lake Formation に登録します。
3. Amazon EMR Studio インスタンスをまだ作成していない場合 (Amazon EMR Serverless を使用するには Amazon EMR Studio が必要です) 。
4. EMR Serverless アプリを作成する
 - リリースバージョン emr-7.7.0 を選択します。
 - ARM64 アーキテクチャを選択します。
 - Opt for カスタム設定を使用します。
 - 事前初期化された容量を無効にします。
 - インタラクティブ作業を行う予定の場合は、インタラクティブエンドポイント > EMR Studio のエンドポイントを有効にするを選択します。
 - 追加設定 > Lake Formation を使用してきめ細かなアクセスコントロールを選択します。
 - アプリケーションの作成
5. EMR-Studio ノートブックまたは StartJobRun API を介して EMR-S を使用します。

分析テンプレートジョブが開始されない

一般的な原因

分析テンプレートジョブは、次の 3 つの主な設定問題により、起動時にすぐに失敗する可能性があります。

- 必要な形式と一致しないスクリプトの命名が正しくない
- ユーザースクリプトにエントリポイント関数がないか、正しくフォーマットされていません
- 仮想環境で互換性のない Python バージョン

解決策

解決するには:

1. ユーザースクリプトを確認します。

- ユーザースクリプトに有効な Python ファイル名があることを確認します。

有効な Python ファイル名では、小文字、アンダースコアを使用して単語を区切り、.py 拡張子を使用します。

2. エントリーポイント関数を確認します。ユーザースクリプトにエントリーポイント関数がない場合は、エントリーポイント関数を追加します。

- a. ユーザースクリプトを開きます。
- b. このエントリーポイント関数を追加します。

```
def entrypoint(context):  
    # Your analysis code here
```

- c. 関数名が `entrypoint` と正確にスペルされていることを確認します。
- d. 関数が `context` パラメータを受け入れていることを確認します。

3. Python バージョンの互換性を確認します。

- a. 仮想環境が Python 3.9 または 3.11 を使用していることを確認します。
- b. バージョンを確認するには、以下を実行します。 `python --version`
- c. 必要に応じて、仮想環境を更新します。

```
conda create -n analysis-env python=3.9  
conda activate analysis-env
```

防止

- 提供された、正しいファイル構造を含む分析テンプレートスターターコードを使用します。
- すべての分析テンプレートに Python 3.9 または 3.11 を使用して専用の仮想環境を設定します。
- ジョブを送信する前に、テンプレート検証ツールを使用して分析テンプレートをローカルでテストします。
- CI/CD チェックを実装して、スクリプトの命名とエントリーポイント関数の要件を検証します。

分析テンプレートジョブは開始されますが、処理中に失敗します

一般的な原因

分析ジョブは、セキュリティ上およびフォーマット上の理由から、実行中に失敗する可能性があります。

- Amazon S3 や などの AWS サービスへの不正な直接アクセスの試み AWS Glue
- 必要な DataFrame 仕様と一致しない誤った形式で出力を返す
- 実行環境のセキュリティ制限によるネットワーク呼び出しのブロック

解決策

解決するには

1. 直接 AWS サービスアクセスを削除します。
 - a. コードで直接 AWS サービスのインポートと呼び出しを検索します。
 - b. 直接 S3 アクセスを提供された Spark セッションメソッドに置き換えます。
 - c. コラボレーションインターフェイスでは、事前設定されたテーブルのみを使用します。
2. 出力を正しくフォーマットします。
 - a. すべての出力が Spark DataFrames であることを確認します。
 - b. 戻りステートメントを次の形式に合わせて更新します。

```
return {  
    "results": {  
        "output1": dataframe1  
    }  
}
```

- c. DataFrame 以外の戻りオブジェクトをすべて削除します。
3. ネットワーク呼び出しを削除します。
 - a. 外部 API コールを特定して削除します。
 - b. urllib、リクエスト、または同様のネットワークライブラリを削除します。
 - c. ソケット接続または HTTP クライアントコードを削除します。

防止

- 提供されたコード linter を使用して、不正な AWS インポートやネットワーク呼び出しがないか確認します。
- セキュリティ制限が本番環境と一致する開発環境でジョブをテストします。
- ジョブをデプロイする前に、出力スキーマの検証プロセスに従います。
- 承認されたサービスアクセスパターンのセキュリティガイドラインを確認してください。

仮想環境の設定が失敗する

一般的な原因

仮想環境設定の失敗は、一般的に以下が原因で発生します。

- 開発環境と実行環境間の CPU アーキテクチャの不一致
- 適切な環境の初期化を妨げる Python コードフォーマットの問題
- コンテナ設定のベースイメージ設定が正しくない

解決策

解決するには

1. 正しいアーキテクチャを設定します。

- a. で現在のアーキテクチャを確認する `uname -m`.
- b. Dockerfile を更新して ARM64 を指定します。

```
FROM --platform=linux/arm64 public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
```

- c. を使用してコンテナを再構築する `docker build --platform=linux/arm64`.

2. Python インデントを修正:

- a. コードファイル `black` で のような Python コードフォーマットを実行します。
- b. スペースまたはタブが一貫して使用されていることを確認します (両方ではない)。
- c. すべてのコードブロックのインデントを確認します。

```
def my_function():
```

```
if condition:
    do_something()
return result
```

- d. Python インデント強調表示で IDE を使用します。
3. 環境設定を検証します。
 - a. を実行して `python -m py_compile your_script.py` 構文エラーを確認します。
 - b. デプロイ前に環境をローカルでテストします。
 - c. すべての依存関係が にリストされていることを確認します `requirements.txt`。

防止

- Python フォーマットプラグインで Visual Studio Code または PyCharm を使用する
- コードフォーマッターを自動的に実行するように事前コミットフックを設定する
- 提供された ARM64 ベースイメージを使用してローカルで環境を構築およびテストする
- CI/CD パイプラインに自動コードスタイルチェックを実装する

コラボレーションでデータを分析する

では AWS Clean Rooms、クエリまたはジョブを実行してデータを分析できます。

クエリは、サポートされている関数、クラス、変数のセットを使用して、コラボレーションの設定済みテーブルにアクセスして分析する方法です。で現在サポートされているクエリ言語 AWS Clean Rooms は SQL です。でクエリを実行するには、SQL コードの AWS Clean Rooms 記述、承認された SQL 分析テンプレートの使用、または Analysis Builder UI の使用の 3 つの方法があります。

ジョブは、サポートされている関数、クラス、変数のセットを使用して、コラボレーションの設定済みテーブルにアクセスして分析する方法です。で現在サポートされているジョブのタイプ AWS Clean Rooms は PySpark です。でジョブを実行する方法は 1 つあります AWS Clean Rooms。承認された PySpark 分析テンプレートを使用する方法です。

以下のトピックでは、SQL クエリまたは PySpark ジョブを実行して AWS Clean Rooms でデータを分析する方法について説明します。

トピック

- [SQL クエリの実行](#)
- [PySpark ジョブの実行](#)

SQL クエリの実行

Note

クエリを実行できるのは、クエリの計算コストを負担するメンバーがアクティブメンバーとしてコラボレーションに参加している場合のみです。

[クエリを実行できるメンバー](#)は、次の方法で SQL クエリを実行できます。

- SQL コードエディタを使用して SQL クエリを手動で構築します。
- 承認された SQL [分析テンプレート](#)を使用する。
- Analysis Builder UI を使用して、SQL コードを記述することなくクエリを構築します。

クエリを実行できるメンバーがコラボレーション内のテーブルに対して SQL クエリを実行すると、は関連するロールを AWS Clean Rooms 引き受けて、ユーザーに代わってテーブルにアクセスします。は、必要に応じて、入力クエリとその出力に分析ルール AWS Clean Rooms を適用します。

分析ルールと出力制約は自動的に適用されます。は、定義された分析ルールに準拠した結果 AWS Clean Rooms のみを返します。

AWS Clean Rooms は、他のクエリエンジンとは異なる SQL クエリをサポートしています。仕様については、「[AWS Clean Rooms SQL リファレンス](#)」を参照してください。差分プライバシーで保護されているデータテーブルでクエリを実行する場合は、クエリが AWS Clean Rooms 差分プライバシーの[汎用クエリ構造](#)と互換性があることを確認する必要があります。

Note

[Cryptographic Computing for Clean Rooms](#) を使用する場合、すべての SQL オペレーションが有効な結果を生成するわけではありません。例えば、暗号化された列に対して COUNT を実行することはできますが、暗号化された数値に対して SUM を実行するとエラーが発生します。また、誤ったクエリ結果が得られることもあります。例えば、シール列に SUM を実行するクエリではエラーが発生します。一方、シール列に対する GROUP BY クエリは成功するように見えますが、クリアテキストに対する GROUP BY クエリで生成されるグループとは異なるグループが生成されます。

[クエリの計算コストを負担するメンバー](#)に、コラボレーションで実行されたクエリの料金が請求されます。

クエリできるメンバーは、[結果を受け取ることができる複数のメンバー](#)を選択して、1 つのクエリから結果を受け取ることができます。詳細については、「[SQL コードエディタを使用して設定済みテーブルのクエリを実行する](#)」を参照してください。クエリ結果の受信に関する一般的な情報については、「」を参照してください[分析結果の受信と使用](#)。

前提条件

SQL クエリを実行する前に、以下があることを確認してください。

- AWS Clean Rooms コラボレーションのアクティブなメンバーシップ
- コラボレーションで少なくとも 1 つの設定済みテーブルへのアクセス
- クエリコンピューティングコストを担当するメンバーがアクティブなコラボレーションメンバーであることの確認

AWS Clean Rooms [StartProtectedQuery API](#) オペレーションを直接呼び出すか、AWS SDKs を使用してデータをクエリまたはクエリを表示する方法については、[AWS Clean Rooms API リファレンス](#)を参照してください。

クエリログ記録の詳細については、「[分析ログイン AWS Clean Rooms](#)」を参照してください。

Note

暗号化されたデータテーブルにクエリを実行すると、暗号化された列の結果は暗号化されます。

SQL クエリの Spark プロパティ設定

AWS Clean Rooms では、SQL クエリでサポートされている Spark プロパティを設定することで、オプションで Spark ランタイムの動作をカスタマイズできます。これらのプロパティを使用すると、パフォーマンス、メモリ使用量、クエリ実行パラメータを微調整できます。この機能を使用すると、Spark ベースのクエリの処理方法をより詳細に制御できるため、特定のワークロード要件に基づいて最適化できます。

シャッフルパーティション、ブロードキャスト結合しきい値、アダプティブクエリ実行パラメータなどの設定を AWS Clean Rooms コンソールから直接調整できるようになりました。この機能は、デフォルト設定が最適でない可能性のある複雑なクエリや大規模なデータセットに特に役立ちます。これらの Spark プロパティを微調整することで、クエリのパフォーマンスを向上させ、リソースの消費量を減らし、Spark ベースのコラボレーション分析のメモリ使用量をより適切に管理できます。

この機能を活用するには、クエリインターフェイスに新しい Spark プロパティセクションがあります。サポートされているプロパティのリストからを選択し、カスタム値を指定できます。[StartProtectedQuery API を使用してプログラムで Spark プロパティを設定することもできます。](#)この高度な設定オプションにより、データアナリストやエンジニアはクエリを最適化して効率とスケーラビリティを向上させることができます。

デフォルト値を含む Spark プロパティの詳細については、Apache Spark [ドキュメントの「Spark プロパティ」](#)を参照してください。

以下のトピックでは、AWS Clean Rooms コンソールを使用してコラボレーション内のデータにクエリを実行する方法を説明します。

トピック

- [SQL コードエディタを使用して設定済みテーブルのクエリを実行する](#)
- [SQL コードエディタを使用して ID マッピングテーブルのクエリを実行する](#)
- [SQL 分析テンプレートを使用した設定済みテーブルのクエリ](#)
- [分析ビルダーを使用してクエリを実行する](#)
- [差分プライバシーの影響を表示する](#)
- [最近のクエリの表示](#)
- [クエリの詳細の表示](#)

SQL コードエディタを使用して設定済みテーブルのクエリを実行する

クエリを行えるメンバーは、SQL コードエディタで SQL コードを記述して手動でクエリを作成できます。SQL コードエディタは、AWS Clean Rooms コンソールの分析タブの分析セクションにあります。

SQL コードエディタはデフォルトで表示されます。分析ビルダーを使用してクエリを作成する場合は、「[分析ビルダーを使用してクエリを実行する](#)」を参照してください。

Important

コードエディタで SQL クエリの作成を開始してから [分析ビルダー UI] をオンにした場合、クエリは保存されません。

AWS Clean Rooms は、多くの SQL コマンド、関数、および条件をサポートしています。詳細については、「[AWS Clean Rooms SQL リファレンス](#)」を参照してください。

Tip

スケジュールされた保守管理がクエリの実行中に発生した場合、クエリは終了し、ロールバックされます。この場合は、クエリをやり直す必要があります。

SQL コードエディタを使用して設定済みテーブルのクエリを実行するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [クエリの実行] になっているコラボレーションを選択します。
4. 分析タブのテーブルで、テーブルのリストと関連する分析ルールタイプ (集計分析ルール、リスト分析ルール、またはカスタム分析ルール) を表示します。

 Note

想定したテーブルが表示されない場合は、次のいずれかの理由が考えられます。

- テーブルが [関連付け](#)られていない。
- テーブルに [分析ルールが設定](#)されていない。

5. (オプション) テーブルのスキーマコントロールと分析ルールコントロールを表示するには、プラス記号アイコン (+) を選択してテーブルを展開します。
6. 分析セクションの分析モードで、SQL コードの書き込み を選択します。

 Note

[分析] セクションは、結果を受け取れるメンバーとクエリの計算コストを負担するメンバーが、アクティブなメンバーとしてコラボレーションに参加している場合にのみ表示されます。

7. SQL コードエディタにクエリを入力してクエリを作成します。

サポートされている SQL コマンドと関数の詳細については、[AWS Clean Rooms 「SQL リファレンス」](#) を参照してください。

次のオプションを使用してクエリを構築することもできます。

Use an example query

サンプルクエリを使用するには

1. テーブルの横にある 3 つの縦のドットを選択します。
2. [エディタに挿入] で [クエリの例] を選択します。

 Note

サンプルクエリを挿入すると、既にエディタにあるクエリに追加されます。

クエリの例が表示されます。[テーブル] の下に表示されているすべてのテーブルがクエリに含まれます。

3. クエリのプレースホルダー値を編集します。

Insert column names or functions

列名または関数を挿入するには

1. 列の横にある 3 つの縦のドットを選択します。
2. [エディタに挿入] で [列名] を選択します。
3. 列で許可されている関数を手動で挿入するには、
 - a. 列の横にある 3 つの縦のドットを選択します。
 - b. エディタで挿入を選択します。
 - c. 許可された関数の名前 (INNER JOIN、SUM DISTINCT、SUM など) を選択します
COUNT。
4. Ctrl + スペースキーを押すと、コードエディタにテーブルスキーマが表示されます。

 Note

クエリを行えるメンバーは、各設定済みテーブルの関連付けのパーティション列を表示して、使用できます。パーティション列が、設定済み AWS Glue テーブルの基盤となるテーブルのパーティション列としてラベル付けされていることを確認します。

5. クエリのプレースホルダー値を編集します。
8. サポートされているワーカータイプとワーカー数を指定します。

SQL クエリを実行するインスタンスタイプとインスタンス数 (ワーカー) を選択できます。

CR.1X では、最大 128 人のワーカーまたは最低 4 人のワーカーを選択できます。

CR.4X では、最大 32 人のワーカーまたは最低 4 人のワーカーを選択できます。

次の表を使用して、ユースケースに必要なタイプと数、またはワーカーを決定します。

ワーカータイプ	vCPU	メモリ (GB)	ストレージ (GB)	ワーカー数	クリーンルーム処理ユニット (CRPU) の合計数
CR.1X (デフォルト)	4	30	100	4	8
				128	256
CR.4X	16	120	400	4	32
				32	256

Note

さまざまなワーカータイプとワーカー数には関連コストがあります。料金の詳細については、「[AWS Clean Rooms の料金](#)」を参照してください。

9. 結果の送信先には、結果を受け取るユーザーを指定します。

Note

結果を受け取るには、コラボレーションメンバーを結果レシーバーとして設定し、コラボレーションのアクティブな参加者である必要があります (ステータス: アクティブ)

10. (クエリのみできるメンバー) デフォルトでは、デフォルトの結果設定を使用するチェックボックスが選択されています。デフォルトの結果設定を維持する場合は、これを選択します。

このクエリに異なる結果設定を指定する場合は、「デフォルトの結果設定を使用する」チェックボックスをオフにしてから、以下を選択します。

- a. 結果形式 (CSV または PARQUET)

- b. 結果ファイル (単一または複数)
- c. Amazon S3 の結果の送信先

結果を受け取ることができる各メンバーは、Amazon S3 で異なる結果形式、結果ファイル、結果の送信先を指定できます。 Amazon S3

11. Spark プロパティを指定するには:

- a. [Spark のプロパティ] を拡張します。
- b. Spark プロパティの追加を選択します。
- c. Spark プロパティダイアログボックスで、ドロップダウンリストからプロパティ名を選択し、値を入力します。

次の表は、各プロパティの定義を示しています。

Spark プロパティの詳細については、Apache [Spark](#) ドキュメントの「Spark Properties」を参照してください。

プロパティ名	説明	デフォルト値
spark.task.maxFailures	ジョブが失敗するまでにタスクが失敗する連続回数を制御します。1 以上の値が必要です。許可される再試行回数は、この値から 1 を引いた値に等しくなります。試行が成功すると、失敗数はリセットされます。さまざまなタスクにまたがる障害は、この制限に累積されません。	4
spark.sql.files.maxPartitionBytes	Parquet、JSON、ORC などのファイルベースのソースから読み取るときに 1 つのパーティションにパックする最大バイト数を設定します。	128MB
spark.hadoop.fs.s3.maxRetries	Amazon S3 ファイルオペレーションの再試行の最大回数を設定します。	

プロパティ名	説明	デフォルト値
spark.network.timeout	すべてのネットワークインタラクションのデフォルトのタイムアウトを設定します。設定されていない場合、次のタイムアウト設定を上書きします。 - spark.storage.blockManagerHeartbeatTimeoutMs - spark.shuffle.io.connectionTimeout - spark.rpc.askTimeout - spark.rpc.lookupTimeout	120 秒
spark.rdd.compress	spark.io.compression.codec を使用してシリアル化された RDD パーティションを圧縮するかどうかを指定します。Java および Scala の StorageLevel.MEMORY_ONLY_SER、または Python の StorageLevel.MEMORY_ONLY に適用されます。ストレージ領域を削減しますが、追加の CPU 処理時間が必要です。	FALSE
spark.shuffle.spill.compress	spark.io.compression.codec を使用してシャッフルスピルデータを圧縮するかどうかを指定します。	TRUE
spark.sql.adaptive.advisoryPartitionSizeInBytes	spark.sql.adaptive.enabled が true の場合、適応最適化中のシャッフルパーティションのターゲットサイズをバイト単位で設定します。小さなパーティションを結合するとき、または歪んだパーティションを分割するときのパーティションサイズを制御します。	(spark.sql.adaptive.shuffle.targetPostShuffleInputSize の値)

プロパティ名	説明	デフォルト値
spark.sql.adaptive.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。適応フレームワークにのみ適用されます。spark.sql.autoBroadcastJoinThreshold と同じデフォルト値を使用します。ブロードキャストを無効にするには、-1 に設定します。	(none)
spark.sql.adaptive.coalescePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytes に基づいて連続したシャッフルパーティションを結合してタスクサイズを最適化するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.coalescePartitions.initialPartitionNum	結合前のシャッフルパーティションの初期数を定義します。spark.sql.adaptive.enabled と spark.sql.adaptive.coalescePartitions.enabled の両方が true である必要があります。デフォルトは spark.sql.shuffle.partitions の値です。	(none)
spark.sql.adaptive.coalescePartitions.minPartitionSize	アダプティブ最適化中にパーティションが小さすぎないように、結合されたシャッフルパーティションの最小サイズを設定します。	1 MB
spark.sql.adaptive.coalescePartitions.parallelismFirst	パーティションの結合中に spark.sql.adaptive.advisoryPartitionSizeInBytes ではなく、クラスターの並列処理に基づいてパーティションサイズを計算するかどうかを指定します。並列処理を最大化するために、設定されたターゲットサイズよりも小さいパーティションサイズを生成します。これをビジー状態のクラスターで false に設定して、過剰な小さなタスクを防止することでリソース使用率を向上させることをお勧めします。	TRUE

プロパティ名	説明	デフォルト値
spark.sql.adaptive.enabled	正確なランタイム統計に基づいて、アダプティブクエリの実行を有効にして、クエリの実行中にクエリプランを再最適化するかどうかを指定します。	TRUE
spark.sql.adaptive.forceOptimizeSkewedJoin	追加のシャッフルを導入した場合でも、OptimizeSkewedJoin を強制的に有効にするかどうかを指定します。	FALSE
spark.sql.adaptive.localShuffleReader.enabled	ソートマージ結合からブロードキャストハッシュ結合に変換した後など、シャッフルパーティショニングが必要ない場合にローカルシャッフルリーダーを使用するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.maxShuffledHashJoinLocalMapThreshold	ローカルハッシュマップを構築するための最大パーティションサイズをバイト単位で設定します。次の場合に、シャッフルされたハッシュ結合をソートマージ結合よりも優先します。 - この値は spark.sql.adaptive.advisoryPartitionSizeInBytes 以上です - すべてのパーティションサイズがこの制限内です spark.sql.join.preferSortMergeJoin 設定を上書きします。	0 バイト

プロパティ名	説明	デフォルト値
spark.sql.adaptive.optimizeSkewsInRebalancePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytes に基づいて小さなパーティションに分割することで、歪んだシャッフルパーティションを最適化するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.rebalancePartitionsSmallPartitionFactor	分割中にパーティションをマージするためのサイズしきい値係数を定義します。この係数より小さいパーティションに spark.sql.adaptive.advisoryPartitionSizeInBytes を掛けるとマージされます。	0.2
spark.sql.adaptive.skewJoin.enabled	スキューされたパーティションを分割してオプションでレプリケートすることで、シャッフルされた結合でデータスキューを処理するかどうかを指定します。ソートマージおよびシャッフルされたハッシュ結合に適用されます。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.skewJoin.skewedPartitionFactor	パーティションスキューを決定するサイズ係数を決定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - この係数にパーティションサイズの中央値を乗算 - spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes の値	5

プロパティ名	説明	デフォルト値
spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes	歪んだパーティションを識別するためのサイズしきい値をバイト単位で設定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - このしきい値 - パーティションサイズの中央値に spark.sql.adaptive.skewJoin.skewedPartitionFactor を掛けた値 この値は spark.sql.adaptive.advisoryPartitionSizeInBytes よりも大きく設定することをお勧めします。	256MB
spark.sql.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。ブロードキャストを無効にするには、-1 に設定します。	10MB
spark.sql.broadcastTimeout	ブロードキャスト結合中のブロードキャストオペレーションのタイムアウト期間を秒単位で制御します。	300 秒
spark.sql.cbo.enabled	計画統計の推定でコストベースの最適化 (CBO) を有効にするかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.star.filter	コストベースの結合列挙中にスター結合フィルターヒューリスティックを適用するかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.threshold	動的プログラミングアルゴリズムで許可される結合ノードの最大数を設定します。	12
spark.sql.cbo.joinReorder.enabled	コストベースの最適化 (CBO) で結合順序変更を有効にするかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.cbo.planStats.enabled	論理計画の生成中にカタログから行数と列統計を取得するかどうかを指定します。	FALSE
spark.sql.cbo.starSchemaDetection	スタースキーマ検出に基づいて結合順序変更を有効にするかどうかを指定します。	FALSE
spark.sql.files.maxPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最大数を設定します。初期数がこの値を超えたときにパーティションを再スケーリングします。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.files.maxRecordsPerFile	1 つのファイルに書き込むレコードの最大数を設定します。ゼロまたは負の値に設定されている場合、制限は適用されません。	0
spark.sql.files.minPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最小数を設定します。デフォルトは spark.sql.leafNodeDefaultParallelism です。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.inMemoryColumnarStorage.batchSize	列キャッシュのバッチサイズを制御します。サイズを大きくすると、メモリ使用率と圧縮が向上しますが、out-of-memoryエラーのリスクが高まります。	10000
spark.sql.inMemoryColumnarStorage.compressed	データ統計に基づいて列の圧縮コーデックを自動的に選択するかどうかを指定します。	TRUE

プロパティ名	説明	デフォルト値
spark.sql.inMemoryColumnarStorage.enableVectorizedReader	列キャッシュのベクトル化された読み取りを有効にするかどうかを指定します。	TRUE
spark.sql.legacy.allowHashOnMapType	マップタイプデータ構造でハッシュオペレーションを許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンのマップタイプの処理との互換性を維持します。	
spark.sql.legacy.allowNegativeScaleOfDecimal	10 進数型定義で負のスケール値を許可するかどうかを指定します。このレガシー設定は、負の 10 進スケールをサポートした古い Spark バージョンとの互換性を維持します。	
spark.sql.legacy.castComplexTypesToString.enabled	複雑な型を文字列にキャストするためのレガシー動作を有効にするかどうかを指定します。古い Spark バージョンのタイプ変換ルールとの互換性を維持します。	
spark.sql.legacy.charVarcharAsString	CHAR 型と VARCHAR 型を STRING 型として扱うかどうかを指定します。このレガシー設定は、古い Spark バージョンの文字列タイプの処理との互換性を提供します。	
spark.sql.legacy.createEmptyCollectionUsingStringType	文字列タイプの要素を使用して空のコレクションを作成するかどうかを指定します。このレガシー設定は、古い Spark バージョンのコレクションの初期化動作との互換性を維持します。	
spark.sql.legacy.exponentLiteralAsDecimal.enabled	指数リテラルを 10 進数型として解釈するかどうかを指定します。このレガシー設定は、古い Spark バージョンの数値リテラル処理との互換性を維持します。	

プロパティ名	説明	デフォルト値
spark.sql.legacy.json.allowEmptyString.enabled	JSON 処理で空の文字列を許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンの JSON 解析動作との互換性を維持します。	
spark.sql.legacy.parquet.int96RebaseModelRead	Parquet ファイルの読み取り時にレガシー INT96 タイムスタンプリベースモードを使用するかどうかを指定します。このレガシー設定は、古い Spark バージョンのタイムスタンプ処理との互換性を維持します。	
spark.sql.legacy.timeParserPolicy	後方互換性のための解析動作を制御します。このレガシー設定により、文字列からタイムスタンプと日付を解析する方法が決まります。	
spark.sql.legacy.typeCoercion.datetimeToString.enabled	日時値を文字列に変換するときにレガシー型の強制動作を有効にするかどうかを指定します。古い Spark バージョンの日時変換ルールとの互換性を維持します。	
spark.sql.maxSinglePartitionBytes	最大パーティションサイズをバイト単位で設定します。プランナーは、並列処理を改善するために、大きなパーティションのシャッフルオペレーションを導入します。	128 メートル

プロパティ名	説明	デフォルト値
spark.sql.metadataCacheTTLSeconds	メタデータキャッシュtime-to-live (TTL) を制御します。パーティションファイルメタデータとセッションカタログキャッシュに適用されます。以下が必要です。 0 より大きい正の値 - spark.sql.catalogImplementationを hive に設定 - spark.sql.hive.filesourcePartitionFileCacheSize が 0 より大きい - spark.sql.hive.manageFilesourcePartitions を true に設定	-1000 ミリ秒
spark.sql.optimizer.collapseProjectAlwaysInline	重複が発生した場合でも、隣接する射影とインライン式を折りたたむかどうかを指定します。	FALSE
spark.sql.optimizer.dynamicPartitionPruning.enabled	結合キーとして使用されるパーティション列の述語を生成するかどうかを指定します。	TRUE
spark.sql.optimizer.enableCsvExpressionOptimization	from_csv オペレーションから不要な列を削除して、SQL オプティマイザの CSV 式を最適化するかどうかを指定します。	TRUE
spark.sql.optimizer.enableJsonExpressionOptimization	SQL オプティマイザで JSON 式を最適化するかどうかを次のように指定します。 - from_json オペレーションから不要な列を削除する - from_json と to_json の組み合わせの簡素化 - named_struct オペレーションの最適化	TRUE

プロパティ名	説明	デフォルト値
<code>spark.sql.optimizer.excludedRules</code>	無効にするオプティマイザルールを定義し、カンマ区切りのルール名で識別します。一部のルールは、正確性のために必要であるため、無効にすることはできません。オプティマイザは、どのルールが正常に無効化されたかをログに記録します。	(none)
<code>spark.sql.optimizer.runtime.bloomFilter.applicationSideScanSizeThreshold</code>	アプリケーション側に Bloom フィルターを挿入するために必要な最小集約スキャンサイズをバイト単位で設定します。	10GB
<code>spark.sql.optimizer.runtime.bloomFilter.creationSideThreshold</code>	作成側で Bloom フィルターを挿入するための最大サイズしきい値を定義します。	10MB
<code>spark.sql.optimizer.runtime.bloomFilter.enabled</code>	シャッフル結合の片側に選択的述語がある場合に、シャッフルデータを減らすために Bloom フィルターを挿入するかどうかを指定します。	TRUE
<code>spark.sql.optimizer.runtime.bloomFilter.expectedNumItems</code>	ランタイム Bloom フィルターで予想される項目のデフォルトの数を定義します。	1000000
<code>spark.sql.optimizer.runtime.bloomFilter.maxNumBits</code>	ランタイム Bloom フィルターで許可される最大ビット数を設定します。	67108864
<code>spark.sql.optimizer.runtime.bloomFilter.maxNumItems</code>	ランタイム Bloom フィルターで許可される予想項目の最大数を設定します。	4000000

プロパティ名	説明	デフォルト値
spark.sql.optimize r.runtime.bloomFilter.number.threshold	ドライバーのout-of-memoryエラーを防ぐために、クエリごとに許可される非 DPP ランタイムフィルターの最大数を制限します。	10
spark.sql.optimize r.runtime.bloomFilter.numBits	ランタイム Bloom フィルターで使用するデフォルトのビット数を定義します。	8388608
spark.sql.optimize r.runtime.rowlevel OperationGroupFilter.enabled	行レベルのオペレーションでランタイムグループのフィルタリングを有効にするかどうかを指定します。データソースに以下を許可します。 - データソースフィルターを使用してデータグループ全体 (ファイルやパーティションなど) を削除する - ランタイムクエリを実行して一致するレコードを特定する - 不要なグループを破棄して、高価な書き換えを避ける 機能制限: - すべての式がデータソースフィルターに変換できるわけではありません - 一部の式では、Spark 評価 (サブクエリなど) が必要です。	TRUE
spark.sql.optimize r.runtimeFilter.semiJoinReduction.enabled	シャッフル結合の片側に選択述語がある場合に、シャッフルデータを減らすためにセミ結合を挿入するかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.parquet.aggregatePushdown	最適化のために集計を Parquet にプッシュダウンするかどうかを指定します。以下をサポートします。 • ブール型、整数型、浮動小数点型、日付型の MIN と MAX • すべてのデータ型の COUNT Parquet ファイルフッターに統計情報がない場合、例外をスローします。	FALSE
spark.sql.parquet.columnarReaderBatchSize	Parquet ベクトル化された各リーダーバッチの行数を制御します。out-of-memoryエラーを防ぐために、パフォーマンスのオーバーヘッドとメモリ使用量のバランスを取る値を選択します。	4096
spark.sql.session.timeZone	文字列リテラルおよび Java オブジェクト変換のタイムスタンプを処理するセッションタイムゾーンを定義します。以下を受け入れます。 • 地域/都市形式のリージョンベースの IDs (アメリカ/Los_Angeles など) • (+/-)HH、(+/-)HH:mm、または (+/-)HH:mm:ss 形式のゾーンオフセット (-08 や +01:00 など) • +00:00 のエイリアスとしての UTC または Z	(ローカルタイムゾーンの値)
spark.sql.shuffle.partitions	結合または集約中のデータシャッフルのパーティションのデフォルト数を設定します。同じチェックポイントの場所からの構造化ストリーミングクエリの再起動の間に変更することはできません。	200

プロパティ名	説明	デフォルト値
spark.sql.shuffledHashJoinFactor	シャッフルハッシュ結合の適格性を判断するために使用される乗算係数を定義します。シャッフルハッシュ結合は、スモールサイドのデータサイズにこの係数を掛けた値がラージサイドのデータサイズより小さい場合に選択されます。	3
spark.sql.sources.parallelPartitionDiscovery.threshold	ファイルベースのソース (Parquet、JSON、ORC) を使用したドライバー側のファイルリストの最大パス数を設定します。パーティション検出中に超過すると、ファイルは別の Spark 分散ジョブを使用して一覧表示されます。	32
spark.sql.statistics.histogram.enabled	推定精度を向上させるために、列統計の計算中に等高ヒストグラムを生成するかどうかを指定します。基本的な列統計に必要な範囲を超える追加のテーブルスキャンが必要です。	FALSE

12. [Run] (実行) を選択します。

Note

結果を受け取れるメンバーがクエリ結果の設定を行っていないと、クエリを実行できません。

13. [結果] を表示します。

詳細については、「[分析結果の受信と使用](#)」を参照してください。

14. 引き続きパラメータを調整してクエリを再度実行するか、[+] ボタンを選択して新しいタブで新しいクエリを開始します。

Note

AWS Clean Rooms は、明確なエラーメッセージを提供することを目指しています。トラブルシューティングに役立つ情報がエラーメッセージに不足している場合は、アカウントチー

ムに連絡し、エラーが発生した経緯とエラーメッセージ (ID を含む) の詳細を伝えてください。詳細については、「[トラブルシューティング AWS Clean Rooms](#)」を参照してください。

SQL コードエディタを使用して ID マッピングテーブルのクエリを実行する

次の手順では、ID マッピングテーブルでマルチテーブル結合クエリを実行して `sourceId` を `targetId` と結合する方法について説明します。

ID マッピングテーブルのクエリを実行する前に、ID マッピングテーブルに正常にデータが入力されている必要があります。

SQL コードエディタを使用して ID マッピングテーブルのクエリを実行するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [クエリの実行] になっているコラボレーションを選択します。
4. 分析タブで、分析セクションに移動します。

Note

[分析] セクションは、結果を受け取れるメンバーとクエリの計算コストを負担するメンバーが、アクティブなメンバーとしてコラボレーションに参加している場合にのみ表示されます。

5. 分析タブの Tables で、ID マッピングテーブル (Managed by AWS Clean Rooms の下) とそれに関連する分析ルールタイプ (ID マッピングテーブル分析ルール) のリストを表示します。

Note

予期している ID マッピングテーブルがリストに表示されない場合は、ID マッピングテーブルに正常にデータが入力されていないことが原因である可能性があります。詳細については、「[既存の ID マッピングテーブルへのデータ入力](#)」を参照してください。

6. SQL コードエディタにクエリを入力してクエリを作成します。

(オプション) クエリの例を使用する場合	(オプション) テーブル名を挿入する場合
1. テーブルの横にある 3 つの縦のドットを選択します。 2. [エディタに挿入] で [JOIN ステートメントの例] を選択します。 Note サンプル JOIN ステートメントを挿入すると、既にエディタにクエリが追加されます。 JOIN ステートメントの例が表示されます。 3. クエリのプレースホルダー値を編集します。	1. 列の横にある 3 つの縦のドットを選択します。 2. [エディタに挿入] で [テーブル名] を選択します。 3. クエリのプレースホルダー値を編集します。

7. サポートされているワーカータイプとワーカー数を指定します。

次の表を使用して、ユースケースに必要なタイプと数、またはワーカーを決定します。

ワーカータイプ	vCPU	メモリ (GB)	ストレージ (GB)	ワーカー数	クリーンルーム処理ユニット (CRPU) の合計
CR.1X (デフォルト)	4	30	100	4	8
				128	256
CR.4X	16	120	400	4	32
				32	256

Note

さまざまなワーカータイプとワーカー数には関連コストがあります。料金の詳細については、「[AWS Clean Rooms の料金](#)」を参照してください。

8. [Run] (実行) を選択します。

Note

結果を受け取れるメンバーがクエリ結果の設定を行っていないと、クエリを実行できません。

9. [結果] を表示します。

詳細については、「[分析結果の受信と使用](#)」を参照してください。

10. 引き続きパラメータを調整してクエリを再度実行するか、[+] ボタンを選択して新しいタブで新しいクエリを開始します。

Note

AWS Clean Rooms は、明確なエラーメッセージを提供することを目指しています。トラブルシューティングに役立つ情報がエラーメッセージに不足している場合は、アカウントチームに連絡し、エラーが発生した経緯とエラーメッセージ (ID を含む) の詳細を伝えてください。詳細については、「[トラブルシューティング AWS Clean Rooms](#)」を参照してください。

SQL 分析テンプレートを使用した設定済みテーブルのクエリ

この手順では、AWS Clean Rooms コンソールで分析テンプレートを使用して、カスタム分析ルールで設定済みテーブルをクエリする方法を示します。

SQL 分析テンプレートを使用してカスタム分析ルールで設定済みテーブルをクエリするには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [クエリの実行] になっているコラボレーションを選択します。
4. 分析タブのテーブルセクションで、テーブルと関連する分析ルールタイプ (カスタム分析ルール) を表示します。

 Note

想定したテーブルが表示されない場合は、次のいずれかの理由が考えられます。

- テーブルが [関連付け](#) られていない。
- テーブルに [分析ルールが設定](#) されていない。

5. 分析セクションの「分析モード」で、「分析テンプレートの実行」を選択し、ドロップダウンリストから分析テンプレートを選択します。
6. SQL 分析テンプレートを形成するパラメータは、定義に自動的に入力されます。
7. サポートされているワーカータイプとワーカー数を指定します。

次の表を使用して、ユースケースに必要なタイプと数、またはワーカーを決定します。

ワーカータイプ	vCPU	メモリ (GB)	ストレージ (GB)	ワーカー数	クリーンルーム処理ユニット (CRPU) の合計
CR.1X (デフォルト)	4	30	100	4	8
				128	256
CR.4X	16	120	400	4	32
				32	256

Note

さまざまなワーカータイプとワーカー数には関連コストがあります。料金の詳細については、「[AWS Clean Rooms の料金](#)」を参照してください。

8. サポートされている Spark プロパティを指定します。
 - a. Spark プロパティの追加を選択します。
 - b. Spark プロパティダイアログボックスで、ドロップダウンリストからプロパティ名を選択し、値を入力します。

次の表は、各プロパティの定義を示しています。

Spark プロパティの詳細については、Apache [Spark](#) ドキュメントの「Spark Properties」を参照してください。

プロパティ名	説明	デフォルト値
spark.task.maxFailures	ジョブが失敗するまでにタスクが失敗する連続回数を制御します。1 以上の値が必要です。許可される再試行回数は、この値から 1 を引いた値に等しくなります。試行が成功すると、失敗回数がリセットされます。さまざまなタスクにまたがる障害は、この制限に累積されません。	4
spark.sql.files.maxPartitionBytes	Parquet、JSON、ORC などのファイルベースのソースから読み取るときに 1 つのパーティションにパックする最大バイト数を設定します。	128MB
spark.hadoop.fs.s3.maxRetries	Amazon S3 ファイルオペレーションの再試行の最大回数を設定します。	
spark.network.timeout	すべてのネットワークインタラクションのデフォルトのタイムアウトを設定します。設定	120 秒

プロパティ名	説明	デフォルト値
	されていない場合、次のタイムアウト設定を上書きします。 - spark.storage.blockManagerHeartbeatTimeoutMs - spark.shuffle.io.connectionTimeout - spark.rpc.askTimeout - spark.rpc.lookupTimeout	
spark.rdd.compress	spark.io.compression.codec を使用してシリアル化された RDD パーティションを圧縮するかどうかを指定します。Java および Scala の StorageLevel.MEMORY_ONLY_SER、または Python の StorageLevel.MEMORY_ONLY に適用されます。ストレージ領域を削減しますが、追加の CPU 処理時間が必要です。	FALSE
spark.shuffle.spill.compress	spark.io.compression.codec を使用してシャッフルスピルデータを圧縮するかどうかを指定します。	TRUE
spark.sql.adaptive.advisoryPartitionSizeInBytes	spark.sql.adaptive.enabled が true の場合、適応最適化中のシャッフルパーティションのターゲットサイズをバイト単位で設定します。小さなパーティションを結合するとき、または歪んだパーティションを分割するときのパーティションサイズを制御します。	(spark.sql.adaptive.shuffle.targetPostShuffleInputSize の値)

プロパティ名	説明	デフォルト値
spark.sql.adaptive.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。適応フレームワークにのみ適用されます。spark.sql.autoBroadcastJoinThresholdと同じデフォルト値を使用します。ブロードキャストを無効にするには、-1に設定します。	(none)
spark.sql.adaptive.coalescePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytesに基づいて連続したシャッフルパーティションを結合してタスクサイズを最適化するかどうかを指定します。spark.sql.adaptive.enabledがtrueである必要があります。	TRUE
spark.sql.adaptive.coalescePartitions.initialPartitionNum	結合前のシャッフルパーティションの初期数を定義します。spark.sql.adaptive.enabledとspark.sql.adaptive.coalescePartitions.enabledの両方がtrueである必要があります。デフォルトはspark.sql.shuffle.partitionsの値です。	(none)
spark.sql.adaptive.coalescePartitions.minPartitionSize	アダプティブ最適化中にパーティションが小さすぎないように、結合されたシャッフルパーティションの最小サイズを設定します。	1 MB
spark.sql.adaptive.coalescePartitions.parallelismFirst	パーティションの結合中にspark.sql.adaptive.advisoryPartitionSizeInBytesではなく、クラスターの並列処理に基づいてパーティションサイズを計算するかどうかを指定します。並列処理を最大化するために、設定されたターゲットサイズよりも小さいパーティションサイズを生成します。これをビジー状態のクラスターでfalseに設定して、過剰な小さなタスクを防止することでリソース使用率を向上させることをお勧めします。	TRUE

プロパティ名	説明	デフォルト値
<code>spark.sql.adaptive.enabled</code>	正確なランタイム統計に基づいて、アダプティブクエリの実行を有効にして、クエリの実行中にクエリプランを再最適化するかどうかを指定します。	TRUE
<code>spark.sql.adaptive.forceOptimizeSkewedJoin</code>	追加のシャッフルを導入した場合でも、 <code>OptimizeSkewedJoin</code> を強制的に有効にするかどうかを指定します。	FALSE
<code>spark.sql.adaptive.localShuffleReader.enabled</code>	ソートマージ結合からブロードキャストハッシュ結合に変換した後など、シャッフルパーティショニングが必要ない場合にローカルシャッフルリーダーを使用するかどうかを指定します。 <code>spark.sql.adaptive.enabled</code> が true である必要があります。	TRUE
<code>spark.sql.adaptive.maxShuffledHashJoinLocalMapThreshold</code>	ローカルハッシュマップを構築するための最大パーティションサイズをバイト単位で設定します。次の場合に、シャッフルされたハッシュ結合をソートマージ結合よりも優先します。 - この値は <code>spark.sql.adaptive.advisoryPartitionSizeInBytes</code> 以上です - すべてのパーティションサイズがこの制限内です <code>spark.sql.join.preferSortMergeJoin</code> 設定を上書きします。	0 バイト

プロパティ名	説明	デフォルト値
spark.sql.adaptive.optimizeSkewsInRebalancePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytes に基づいて小さなパーティションに分割することで、歪んだシャッフルパーティションを最適化するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.rebalancePartitionsSmallPartitionFactor	分割中にパーティションをマージするためのサイズしきい値係数を定義します。この係数より小さいパーティションに spark.sql.adaptive.advisoryPartitionSizeInBytes を掛けるとマージされます。	0.2
spark.sql.adaptive.skewJoin.enabled	スキューされたパーティションを分割してオプションでレプリケートすることで、シャッフルされた結合でデータスキューを処理するかどうかを指定します。ソートマージおよびシャッフルされたハッシュ結合に適用されます。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.skewJoin.skewedPartitionFactor	パーティションスキューを決定するサイズ係数を決定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - この係数にパーティションサイズの中央値を乗算 - spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes の値	5

プロパティ名	説明	デフォルト値
spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes	歪んだパーティションを識別するためのサイズしきい値をバイト単位で設定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - このしきい値 - パーティションサイズの中央値に spark.sql.adaptive.skewJoin.skewedPartitionFactor を掛けた値 この値は spark.sql.adaptive.advisoryPartitionSizeInBytes よりも大きく設定することをお勧めします。	256MB
spark.sql.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。ブロードキャストを無効にするには、-1 に設定します。	10MB
spark.sql.broadcastTimeout	ブロードキャスト結合中のブロードキャストオペレーションのタイムアウト期間を秒単位で制御します。	300 秒
spark.sql.cbo.enabled	計画統計の推定でコストベースの最適化 (CBO) を有効にするかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.star.filter	コストベースの結合列挙中にスター結合フィルターヒューリスティックを適用するかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.threshold	動的プログラミングアルゴリズムで許可される結合ノードの最大数を設定します。	12
spark.sql.cbo.joinReorder.enabled	コストベースの最適化 (CBO) で結合順序変更を有効にするかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.cbo.planStats.enabled	論理計画の生成中にカタログから行数と列統計を取得するかどうかを指定します。	FALSE
spark.sql.cbo.starSchemaDetection	スタースキーマ検出に基づいて結合順序変更を有効にするかどうかを指定します。	FALSE
spark.sql.files.maxPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最大数を設定します。初期数がこの値を超えたときにパーティションを再スケールリングします。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.files.maxRecordsPerFile	1 つのファイルに書き込むレコードの最大数を設定します。ゼロまたは負の値に設定されている場合、制限は適用されません。	0
spark.sql.files.minPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最小数を設定します。デフォルトは spark.sql.leafNodeDefaultParallelism です。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.inMemoryColumnarStorage.batchSize	列キャッシュのバッチサイズを制御します。サイズを大きくすると、メモリ使用率と圧縮が向上しますが、out-of-memoryエラーのリスクが高まります。	10000
spark.sql.inMemoryColumnarStorage.compressed	データ統計に基づいて列の圧縮コーデックを自動的に選択するかどうかを指定します。	TRUE

プロパティ名	説明	デフォルト値
spark.sql.inMemoryColumnarStorage.enableVectorizedReader	列キャッシュのベクトル化された読み取りを有効にするかどうかを指定します。	TRUE
spark.sql.legacy.allowHashOnMapType	マップタイプデータ構造でハッシュオペレーションを許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンのマップタイプの処理との互換性を維持します。	
spark.sql.legacy.allowNegativeScaleOfDecimal	10 進数型定義で負のスケール値を許可するかどうかを指定します。このレガシー設定は、負の 10 進スケールをサポートした古い Spark バージョンとの互換性を維持します。	
spark.sql.legacy.castComplexTypesToString.enabled	複雑な型を文字列にキャストするためのレガシー動作を有効にするかどうかを指定します。古い Spark バージョンのタイプ変換ルールとの互換性を維持します。	
spark.sql.legacy.charVarcharAsString	CHAR 型と VARCHAR 型を STRING 型として扱うかどうかを指定します。このレガシー設定は、古い Spark バージョンの文字列タイプの処理との互換性を提供します。	
spark.sql.legacy.createEmptyCollectionUsingStringType	文字列タイプの要素を使用して空のコレクションを作成するかどうかを指定します。このレガシー設定は、古い Spark バージョンのコレクションの初期化動作との互換性を維持します。	
spark.sql.legacy.exponentLiteralAsDecimal.enabled	指数リテラルを 10 進数型として解釈するかどうかを指定します。このレガシー設定は、古い Spark バージョンの数値リテラル処理との互換性を維持します。	

プロパティ名	説明	デフォルト値
spark.sql.legacy.json.allowEmptyString.enabled	JSON 処理で空の文字列を許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンの JSON 解析動作との互換性を維持します。	
spark.sql.legacy.parquet.int96RebaseModelRead	Parquet ファイルの読み取り時にレガシー INT96 タイムスタンプリベースモードを使用するかどうかを指定します。このレガシー設定は、古い Spark バージョンのタイムスタンプ処理との互換性を維持します。	
spark.sql.legacy.timeParserPolicy	後方互換性のための解析動作を制御します。このレガシー設定により、文字列からタイムスタンプと日付を解析する方法が決まります。	
spark.sql.legacy.typeCoercion.datetimeToString.enabled	日時値を文字列に変換するときにレガシー型の強制動作を有効にするかどうかを指定します。古い Spark バージョンの日時変換ルールとの互換性を維持します。	
spark.sql.maxSinglePartitionBytes	最大パーティションサイズをバイト単位で設定します。プランナーは、並列処理を改善するために、大きなパーティションのシャッフルオペレーションを導入します。	128 メートル

プロパティ名	説明	デフォルト値
spark.sql.metadataCacheTTLSeconds	メタデータキャッシュtime-to-live (TTL) を制御します。パーティションファイルメタデータとセッションカタログキャッシュに適用されます。以下が必要です。 0 より大きい正の値 - spark.sql.catalogImplementationを hive に設定 - spark.sql.hive.filesourcePartitionFileCacheSize が 0 より大きい - spark.sql.hive.manageFilesourcePartitions を true に設定	-1000 ミリ秒
spark.sql.optimizer.collapseProjectAlwaysInline	重複が発生した場合でも、隣接する射影とインライン式を折りたたむかどうかを指定します。	FALSE
spark.sql.optimizer.dynamicPartitionPruning.enabled	結合キーとして使用されるパーティション列の述語を生成するかどうかを指定します。	TRUE
spark.sql.optimizer.enableCsvExpressionOptimization	from_csv オペレーションから不要な列を削除して、SQL オプティマイザの CSV 式を最適化するかどうかを指定します。	TRUE
spark.sql.optimizer.enableJsonExpressionOptimization	SQL オプティマイザで JSON 式を最適化するかどうかを次のように指定します。 - from_json オペレーションから不要な列を削除する - from_json と to_json の組み合わせの簡素化 - named_struct オペレーションの最適化	TRUE

プロパティ名	説明	デフォルト値
spark.sql.optimizer.excludedRules	無効にするオプティマイザルールを定義し、カンマ区切りのルール名で識別します。一部のルールは、正確性のために必要であるため、無効にすることはできません。オプティマイザは、正常に無効化されたルールをログに記録します。	(none)
spark.sql.optimizer.runtime.bloomFilter.applicationSideScanSizeThreshold	アプリケーション側に Bloom フィルターを挿入するために必要な最小集約スキャンサイズをバイト単位で設定します。	10GB
spark.sql.optimizer.runtime.bloomFilter.creationSideThreshold	作成側で Bloom フィルターを挿入するための最大サイズしきい値を定義します。	10MB
spark.sql.optimizer.runtime.bloomFilter.enabled	シャッフル結合の片側に選択的述語がある場合に、シャッフルデータを減らすために Bloom フィルターを挿入するかどうかを指定します。	TRUE
spark.sql.optimizer.runtime.bloomFilter.expectedNumItems	ランタイム Bloom フィルターで予想される項目のデフォルトの数を定義します。	1000000
spark.sql.optimizer.runtime.bloomFilter.maxNumBits	ランタイム Bloom フィルターで許可される最大ビット数を設定します。	67108864
spark.sql.optimizer.runtime.bloomFilter.maxNumItems	ランタイム Bloom フィルターで許可される予想項目の最大数を設定します。	4000000

プロパティ名	説明	デフォルト値
spark.sql.optimize r.runtime.bloomFilter.number.threshold	ドライバーのout-of-memoryエラーを防ぐために、クエリごとに許可される非 DPP ランタイムフィルターの最大数を制限します。	10
spark.sql.optimize r.runtime.bloomFilter.numBits	ランタイム Bloom フィルターで使用するデフォルトのビット数を定義します。	8388608
spark.sql.optimize r.runtime.rowlevel OperationGroupFilter.enabled	行レベルのオペレーションでランタイムグループのフィルタリングを有効にするかどうかを指定します。データソースに以下を許可します。 - データソースフィルターを使用してデータグループ全体 (ファイルやパーティションなど) を削除する - ランタイムクエリを実行して一致するレコードを特定する - 不要なグループを破棄して、高価な書き換えを避ける 機能制限: - すべての式がデータソースフィルターに変換できるわけではありません - 一部の式では Spark 評価 (サブクエリなど) が必要です	TRUE
spark.sql.optimize r.runtimeFilter.semiJoinReduction.enabled	シャッフル結合の片側に選択的述語がある場合に、シャッフルデータを減らすためにセミ結合を挿入するかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.parquet.aggregatePushdown	最適化のために集計を Parquet にプッシュダウンするかどうかを指定します。以下をサポートします。 • ブール型、整数型、浮動小数点型、日付型の MIN と MAX • すべてのデータ型の COUNT Parquet ファイルフッターに統計がない場合、例外をスローします。	FALSE
spark.sql.parquet.columnarReaderBatchSize	Parquet ベクトル化された各リーダーバッチの行数を制御します。out-of-memoryエラーを防ぐために、パフォーマンスのオーバーヘッドとメモリ使用量のバランスを取る値を選択します。	4096
spark.sql.session.timeZone	文字列リテラルのタイムスタンプと Java オブジェクト変換を処理するセッションタイムゾーンを定義します。以下を受け入れます。 • 地域/都市形式のリージョンベースの IDs (アメリカ/Los_Angeles など) • (+/-)HH、(+/-)HH:mm、または (+/-)HH:mm:ss 形式のゾーンオフセット (-08 や +01:00 など) • +00:00 のエイリアスとしての UTC または Z	(ローカルタイムゾーンの値)
spark.sql.shuffle.partitions	結合または集約中のデータシャッフルのパーティションのデフォルト数を設定します。同じチェックポイントの場所からの構造化ストリーミングクエリの再起動の間に変更することはできません。	200

プロパティ名	説明	デフォルト値
spark.sql.shuffledHashJoinFactor	シャッフルハッシュ結合の適格性を判断するために使用される乗算係数を定義します。シャッフルハッシュ結合は、スモールサイドのデータサイズにこの係数を掛けた値がラージサイドのデータサイズより小さい場合に選択されます。	3
spark.sql.sources.parallelPartitionDiscovery.threshold	ファイルベースのソース (Parquet、JSON、ORC) を使用したドライバー側のファイルリストの最大パス数を設定します。パーティション検出中に超過すると、ファイルは別の Spark 分散ジョブを使用して一覧表示されます。	32
spark.sql.statistics.histogram.enabled	推定精度を向上させるために、列統計の計算中に等高ヒストグラムを生成するかどうかを指定します。基本的な列統計に必要な範囲を超える追加のテーブルスキャンが必要です。	FALSE

9. [Run] (実行) を選択します。

Note

結果を受け取れるメンバーがクエリ結果の設定を行っていないと、クエリを実行できません。

10. 引き続きパラメータを調整してクエリを再度実行するか、[+] ボタンを選択して新しいタブで新しいクエリを開始します。

分析ビルダーを使用してクエリを実行する

分析ビルダーを使用すると、SQL コードを記述しなくてもクエリを作成できます。分析ビルダーでは、次のようなテーブルが含まれるコラボレーションのクエリを作成できます。

- JOIN を必要とせず、[集計分析ルール](#)を使用する 1 つのテーブル
- [集計分析ルール](#)を両方で使用する 2 つのテーブル (各メンバーから 1 つ)

- [リスト分析ルール](#)を両方で使用する 2 つのテーブル (各メンバーから 1 つ)
- 集計分析ルールを両方で使用する 2 つのテーブル (各メンバーから 1 つ) と、リスト分析ルールを両方で使用する 2 つのテーブル (各メンバーから 1 つ)

SQL クエリを手動で記述する場合は、「[SQL コードエディタを使用して設定済みテーブルのクエリを実行する](#)」を参照してください。

Analysis Builder は、AWS Clean Rooms コンソールの Analysis タブの Analysis セクションに Analysis Builder UI オプションとして表示されます。

 Important

[分析ビルダー UI] をオンにし、分析ビルダーでクエリの作成を開始してから、[分析ビルダー UI] をオフにした場合、クエリは保存されません。

 Tip

スケジュールされた保守管理がクエリの実行中に発生した場合、クエリは終了し、ロールバックされます。この場合は、クエリをやり直す必要があります。

以下のトピックでは、分析ビルダーの使用方法について説明します。

トピック

- [分析ビルダーを使用して 1 つのテーブルにクエリを実行する \(集計\)](#)
- [分析ビルダーを使用して 2 つのテーブルにクエリを実行する \(集計またはリスト\)](#)

分析ビルダーを使用して 1 つのテーブルにクエリを実行する (集計)

この手順では、AWS Clean Rooms コンソールで Analysis Builder UI を使用してクエリを構築する方法を示します。このクエリは、JOIN を必要とせず、[集計分析ルール](#)を使用するテーブルが 1 つだけ含まれるコラボレーション用です。

分析ビルダーを使用して 1 つのテーブルにクエリを実行するには

にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

1. 左のナビゲーションペインで、[コラボレーション] を選択します。
2. [自身のメンバー能力] のステータスが [クエリの実行] になっているコラボレーションを選択します。
3. 分析タブのテーブルで、テーブルと関連する分析ルールタイプを表示します。(分析ルールの種類は集計分析ルールでなければなりません)。

 Note

想定したテーブルが表示されない場合は、次のいずれかの理由が考えられます。

- テーブルが [関連付け](#) られていない。
- テーブルに [分析ルールが設定](#) されていない。

4. [分析] セクションで、[分析ビルダー UI] をオンにします。
5. クエリを作成します。

集計メトリクスをすべて表示する場合は、ステップ 9 に進んでください。

- a. [メトリクスを選択] で、デフォルトで事前に選択されている集計メトリクスを確認し、必要に応じてメトリクスを削除します。
- b. (オプション) [セグメントを追加 – オプション] で、1 つ以上のパラメータを選択します。

 Note

[セグメントを追加 – オプション] はテーブルにディメンションが指定されている場合にのみ表示されます。

- c. (オプション) [フィルターを追加 – オプション] で [フィルターを追加] を選択し、パラメータ、演算子、および値を選択します。

さらにフィルターを追加するには、[別のフィルターを追加] を選択します。

フィルターを削除するには、[削除] を選択します。

 Note

集計クエリでは ORDER BY はサポートされていません。
AND 演算子のみがフィルターでサポートされています。

- d. (オプション) [説明を追加 – オプション] に、クエリのリストでクエリを識別しやすくするための説明を入力します。
6. [SQL コードをプレビュー] を展開します。
 - a. 分析ビルダーから生成された SQL コードを表示します。
 - b. SQL コードをコピーするには、[コピー] を選択します。
 - c. SQL コードを編集するには、[SQL コードエディタで編集] を選択します。
 7. [Run] (実行) を選択します。

 Note

結果を受け取れるメンバーがクエリ結果の設定を行っていないと、クエリを実行できません。

8. 引き続きパラメータを調整してクエリを再度実行するか、[+] ボタンを選択して新しいタブで新しいクエリを開始します。

 Note

AWS Clean Rooms は、明確なエラーメッセージを提供することを目指しています。トラブルシューティングに役立つ情報がエラーメッセージに不足している場合は、アカウントチームに連絡し、エラーが発生した経緯とエラーメッセージ (ID を含む) の詳細を伝えてください。詳細については、「[トラブルシューティング AWS Clean Rooms](#)」を参照してください。

分析ビルダーを使用して 2 つのテーブルにクエリを実行する (集計またはリスト)

この手順では、AWS Clean Rooms コンソールで Analysis Builder を使用して、以下を含むコラボレーションのクエリを構築する方法について説明します。

- [集計分析ルール](#)を両方で使用する 2 つのテーブル (各メンバーから 1 つ)
- [リスト分析ルール](#)を両方で使用する 2 つのテーブル (各メンバーから 1 つ)
- 集計分析ルールを両方で使用する 2 つのテーブル (各メンバーから 1 つ) と、リスト分析ルールを両方で使用する 2 つのテーブル (各メンバーから 1 つ)

分析ビルダーを使用して 2 つのテーブルにクエリを実行する

にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

1. 左のナビゲーションペインで、[コラボレーション] を選択します。
2. メンバーの能力ステータスが Run クエリであるコラボレーションを選択します。
3. 分析タブのテーブルで、2 つのテーブルと関連する分析ルールタイプ (集計分析ルールまたはリスト分析ルール) を表示します。

Note

想定したテーブルが表示されない場合は、次のいずれかの理由が考えられます。

- テーブルが [関連付け](#)られていない。
- テーブルに [分析ルールが設定](#)されていない。

4. [分析] セクションで、[分析ビルダー UI] をオンにします。
5. クエリを作成します。

コラボレーションに、集計分析ルールを使用する 2 つのテーブルと、リスト分析ルールを使用する 2 つのテーブルが含まれている場合は、まず [集約] または [リスト] のどちらかを選択し、選択した分析ルールに基づいてプロンプトに従います。

2 つのテーブルが集計分析ルールを使用している場合

1. [メトリクスを選択] で、デフォルトで事前に選択されている集計メトリクスを確認し、必要に応じてメトリクスを削除します。
2. [レコードを照合] で、1 つ以上のレコードを選択します。

2 つのテーブルがリスト分析ルールを使用している場合

1. [属性を選択] で、デフォルトで事前に選択されているリスト属性を確認し、必要に応じてメトリクスを削除します。
2. [レコードを照合] で、1 つ以上のレコードを選択します。

Note

分析ビルダーを使用する場合、1 組の列でのみ照合できます。

2 つのテーブルが集計分析ルールを使用している場合

Note

分析ビルダーを使用する場合、1 組の列でのみ照合できます。

3. (オプション) [セグメントを追加 – オプション] で、1 つ以上のパラメータを選択します。

Note

[セグメントを追加 – オプション] はテーブルにディメンションが指定されている場合にのみ表示されます。

4. (オプション) [フィルターを追加 – オプション] で [フィルターを追加] を選択し、パラメータ、演算子、および値を選択します。

さらにフィルターを追加するには、[別のフィルターを追加] を選択します。

フィルターを削除するには、[削除] を選択します。

Note

集約クエリでは ORDER BY はサポートされていません。
AND 演算子のみがフィルターでサポートされています。

2 つのテーブルがリスト分析ルールを使用している場合

3. (オプション) [フィルターを追加 – オプション] で [フィルターを追加] を選択し、パラメータ、演算子、および値を選択します。

さらにフィルターを追加するには、[別のフィルターを追加] を選択します。

フィルターを削除するには、[削除] を選択します。

Note

リストクエリでは、LIMIT はサポートされていません。
AND 演算子のみがフィルターでサポートされています。

4. (オプション) [説明を追加 – オプション] に、最近のクエリのリストでクエリを識別しやすくするための説明を入力します。

2 つのテーブルが集計分析ルールを使用している場合

2 つのテーブルがリスト分析ルールを使用している場合

5. (オプション) [説明を追加 – オプション] に、最近のクエリのリストでクエリを識別しやすくするための説明を入力します。

6. [SQL コードをプレビュー] を展開します。
 - a. 分析ビルダーから生成された SQL コードを表示します。
 - b. SQL コードをコピーするには、[コピー] を選択します。
 - c. SQL コードを編集するには、[SQL コードエディタで編集] を選択します。
7. [Run] (実行) を選択します。

 Note

結果を受け取れるメンバーがクエリ結果の設定を行っていないと、クエリを実行できません。

8. 引き続きパラメータを調整してクエリを再度実行するか、[+] ボタンを選択して新しいタブで新しいクエリを開始します。

 Note

AWS Clean Rooms は、明確なエラーメッセージを提供することを目指しています。トラブルシューティングに役立つ情報がエラーメッセージに不足している場合は、アカウントチームに連絡し、エラーが発生した経緯とエラーメッセージ (ID を含む) の詳細を伝えてください。詳細については、「[トラブルシューティング AWS Clean Rooms](#)」を参照してください。

差分プライバシーの影響を表示する

一般に、差分プライバシーが有効になっていても、クエリの記述と実行は変わりません。ただし、プライバシー予算が十分に残っていないと、クエリを実行できません。クエリを実行してプライバシー予算を消費すると、実行可能な集計のおおよその数と、それが将来のクエリにどのような影響を与えるかがわかります。

コラボレーションにおける差分プライバシーの影響を調べるには

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [メンバー詳細] のステータスが [クエリの実行] になっているコラボレーションを選択します。
4. 分析タブのテーブルで、残りのプライバシー予算を表示します。これは、[残りの集計関数] の推定数と [使用されたユーティリティ] (パーセンテージで表示) として表示されます。

Note

[残りの集計関数] の推定数と [使用されたユーティリティ] の割合は、クエリを実行できるメンバーにのみ表示されます。

5. [影響を表示] を選択すると、結果にどの程度のノイズが加わり、実行できる集計関数のおおよその数が表示されます。

最近のクエリの表示

過去 90 日間に実行されたクエリは、分析タブで表示できます。

Note

唯一のメンバー機能がデータ寄与であり、[クエリコンピューティングコストを支払うメンバー](#)ではない場合、分析タブはコンソールに表示されません。

最近のクエリを表示するには

にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

1. 左のナビゲーションペインで、[コラボレーション] を選択します。
2. コラボレーションを選択します。
3. 分析タブの「分析」で、ドロップダウンからすべてのクエリを選択し、過去 90 日間に実行されたクエリを表示します。

4. 最近のクエリをステータス別にソートするには、[すべてのステータス] ドロップダウンリストからステータスを選択します。

ステータスには、[送信済み]、[開始]、[キャンセル済み]、[成功]、[失敗]、[タイムアウト] があります。

クエリの詳細の表示

クエリの詳細は、クエリを行えるメンバーとして、または結果を受け取れるメンバーとして表示できます。

クエリの詳細を表示するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. 分析タブで、次のいずれかを実行します。

- 表示する特定のクエリのオプションボタンを選択して、[詳細を表示] を選択します。
- [保護されたクエリ ID] を選択します。

5. [クエリの詳細] ページで次の操作を行います。

- クエリを行えるメンバーの場合は、[クエリの詳細]、[SQL テキスト]、および [結果] を確認できます。

結果を受け取れるメンバーにクエリ結果が送信されたことを通知するメッセージが表示されます。

- 結果を受け取れるメンバーの場合は、[クエリの詳細] と [結果] を確認できます。

PySpark ジョブの実行

[クエリを実行できるメンバー](#)は、承認された PySpark [分析テンプレート](#) を使用して、設定されたテーブルで PySpark ジョブを実行できます。

前提条件

PySpark ジョブを実行する前に、以下が必要です。

- AWS Clean Rooms コラボレーションのアクティブなメンバーシップ
- コラボレーション内の少なくとも 1 つの分析テンプレートへのアクセス
- コラボレーションで少なくとも 1 つの設定済みテーブルへのアクセス
- 指定された S3 バケットに PySpark ジョブの結果を書き込むアクセス許可

必要なサービスロールの作成については、「」を参照してください[PySpark ジョブの結果を書き込むサービスロールを作成する](#)。

- コンピューティングコストの支払いを担当するメンバーが、アクティブなメンバーとしてコラボレーションに参加しました

StartProtectedJob API オペレーションを直接呼び出す AWS Clean Rooms か、AWS SDKs」を参照してください。 [AWS Clean Rooms](#)

ジョブのログ記録の詳細については、「」を参照してください[分析ログイン AWS Clean Rooms](#)。

ジョブ結果の受信については、「」を参照してください[分析結果の受信と使用](#)。

以下のトピックでは、AWS Clean Rooms コンソールを使用してコラボレーションの設定済みテーブルで PySpark ジョブを実行する方法について説明します。

トピック

- [PySpark 分析テンプレートを使用して設定済みテーブルで PySpark ジョブを実行する](#)
- [最近のジョブの表示](#)
- [ジョブの詳細の表示](#)

PySpark 分析テンプレートを使用して設定済みテーブルで PySpark ジョブを実行する

この手順では、コンソールで AWS Clean Rooms PySpark 分析テンプレートを使用して、カスタム分析ルールで設定済みテーブルを分析する方法を示します。

PySpark 分析テンプレートを使用して設定済みテーブルで PySpark ジョブを実行するには

にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。

1. 左のナビゲーションペインで、[コラボレーション] を選択します。

2. メンバーの能力ステータスが Run jobs のコラボレーションを選択します。
3. 分析タブのテーブルセクションで、テーブルと関連する分析ルールタイプ (カスタム分析ルール) を表示します。

 Note

想定したテーブルが表示されない場合は、次のいずれかの理由が考えられます。

- テーブルが[関連付け](#)られていない。
- テーブルに[分析ルールが設定](#)されていない。

4. 分析セクションの分析モードで、分析テンプレートの実行を選択します。
5. 分析テンプレートドロップダウンリストから PySpark 分析テンプレートを選択します。

PySpark 分析テンプレートのパラメータは、定義に自動的に入力されます。

6. 分析テンプレートにパラメータが定義されている場合は、Parameters でパラメータの値を指定します。
 - a. パラメータごとに、パラメータ名とデフォルト値 (設定されている場合) を表示します。
 - b. 上書きする各パラメータの値を入力します。

 Note

値を指定せず、デフォルト値が存在する場合は、デフォルト値が使用されます。

 Important

パラメータ値は最大 1,000 文字で、UTF-8 エンコーディングをサポートします。すべてのパラメータ値は文字列として扱われ、コンテキストオブジェクトを介してユーザースク립トに渡されます。

ユーザースク립トがパラメータ値を安全に検証して処理していることを確認します。安全なパラメータ処理の詳細については、「」を参照してください[PySpark 分析テンプレートでのパラメータの操作](#)。

7. サポートされているワーカータイプとワーカー数を指定します。

次の表を使用して、ユースケースに必要なタイプと数、またはワーカーを決定します。

ワーカータイプ	vCPU	メモリ (GB)	ストレージ (GB)	ワーカー数	クリーンルーム処理ユニット (CRPU) の合計
CR.1X (デフォルト)	4	30	100	4	8
				128	256
CR.4X	16	120	400	4	32
				32	256

Note

さまざまなワーカータイプとワーカー数には関連コストがあります。料金の詳細については、「[AWS Clean Rooms の料金](#)」を参照してください。

8. [Run] (実行) を選択します。

Note

結果を受け取ることができるメンバーがジョブ結果の設定を行っていない場合、ジョブを実行することはできません。

9. パラメータを調整してジョブを再度実行するか、+ ボタンを選択して新しいタブで新しいジョブを開始します。

最近のジョブの表示

過去 90 日間に実行されたジョブは、分析タブで表示できます。

Note

メンバーの唯一の機能がデータ貢献であり、[ジョブのコンピューティングコストを支払うメンバー](#)ではない場合、分析タブはコンソールに表示されません。

最近のジョブを表示するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. 分析タブの分析で、ドロップダウンからすべてのジョブを選択し、過去 90 日間に実行されたジョブを表示します。
5. 最近のジョブをステータスでソートするには、すべてのステータスドロップダウンリストからステータスを選択します。

ステータスには、[送信済み]、[開始]、[キャンセル済み]、[成功]、[失敗]、[タイムアウト] があります。

ジョブの詳細の表示

ジョブの詳細を、ジョブを実行できるメンバーとして、または結果を受信できるメンバーとして表示できます。

ジョブの詳細を表示するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションを選択します。
4. 分析タブの分析で、ドロップダウンからすべてのジョブを選択し、次のいずれかを実行します。
 - 表示する特定のジョブのオプションボタンを選択し、詳細の表示を選択します。
 - 保護されたジョブ ID を選択します。
5. ジョブの詳細ページで、

- ジョブを実行できるメンバーの場合は、ジョブの詳細、ジョブ、結果を表示します。

ジョブ結果が結果を受け取ることができるメンバーに配信されたことを確認するメッセージが表示されます。

- 結果を受け取ることができるメンバーの場合は、ジョブの詳細と結果を表示します。

分析結果の受信と使用

[結果を受け取ることができるメンバー](#)は、AWS Clean Rooms コンソールまたはコラボレーションに参加したときに指定した Amazon S3 バケットでクエリ結果を確認できます。

Note

暗号化されたデータテーブルの場合のみ、結果を受け取ることができるメンバーが、C3R 暗号化クライアントを[復号](#)モードで実行してクエリ結果を復号します。

Amazon S3 の結果の送信先は、データソースと同じ S3 バケット内に存在することはできません。

以下のトピックでは、AWS Clean Rooms コンソールを使用して分析結果を受信する方法について説明します。

トピック

- [クエリ結果の受信](#)
- [ジョブ結果の受信](#)
- [クエリ結果設定のデフォルト値の編集](#)
- [ジョブ結果設定のデフォルト値の編集](#)
- [他のでのクエリ出力の使用 AWS のサービス](#)

AWS Clean Rooms API を直接呼び出すか、AWS SDKs」を参照してください。 [AWS Clean Rooms](#)

クエリログ記録の詳細については、「[分析ログイン AWS Clean Rooms](#)」を参照してください。

Note

暗号化されたデータテーブルにクエリを実行すると、暗号化された列の結果は暗号化されます。

クエリ結果の受信

Note

Amazon S3 の結果の送信先は、データソースと同じ S3 バケット内に存在することはできません。

クエリの結果は、AWS Clean Rooms コンソールの分析タブの結果設定のデフォルトセクションにあります。

クエリ結果を受け取るには

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [結果を受け取る] になっているコラボレーションを選択します。
4. クエリ結果を から直接受け取るには AWS Clean Rooms、分析タブの分析で、ドロップダウンからすべてのクエリを選択し、保護されたクエリ ID 列でクエリを選択します。
5. [クエリの詳細] ページの [結果] で、次のいずれかを実行します。

目的	選択内容
結果をコピーする。	[Copy] (コピー)
結果をダウンロードする。	ダウンロード
	Note デフォルトでは、ダウンロードされたファイルの名前は、AWS Clean Roomsでクエリが実行されたときに表示された Query id と同じものになります。
Amazon S3 で結果を表示する。	Amazon S3 で表示

目的	選択内容
	別のタブで IAM コンソールが開きます。

6. 暗号化されたデータを使用している場合は、ここでデータテーブルを[復号化](#)できます。

詳細については、「[C3R 暗号化クライアントによるデータテーブルの復号化](#)」を参照してください。

ジョブ結果の受信

Note

Amazon S3 の結果の送信先は、データソースと同じ S3 バケット内に存在することはできません。

ジョブの結果は、AWS Clean Rooms コンソールの分析タブの結果設定のデフォルトセクションにあります。

ジョブ結果を受け取るには

1. にサインイン AWS マネジメントコンソール し、で[AWS Clean Rooms コンソール](#)を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [結果を受け取る] になっているコラボレーションを選択します。
4. ジョブの結果を直接受け取るには AWS Clean Rooms、分析タブの分析で、ドロップダウンからすべてのジョブを選択し、保護対象のジョブ ID 列でジョブを選択します。
5. ジョブの詳細ページの「結果」で、ジョブ ID をコピーします。

分析タブに戻り、結果設定のデフォルトを展開します。

結果の送信先で、リンクを選択して Amazon S3 の結果を表示します。

別のタブで IAM コンソールが開きます。

Amazon S3 で、検索バーにジョブ ID を貼り付け、Enter キーを押します。

結果を含むフォルダが表示されます。フォルダを選択してジョブ結果を表示します。

クエリ結果設定のデフォルト値の編集

Note

Amazon S3 の結果の送信先は、データソースと同じ S3 バケット内に存在することはできません。

結果を受け取ることができるメンバーは、AWS Clean Rooms コンソールでクエリ結果設定のデフォルト値を編集できます。

クエリ結果設定のデフォルト値を編集するには

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [結果を受け取る] になっているコラボレーションを選択します。
4. 分析タブの結果設定のデフォルトで、編集を選択します。
5. 結果設定のデフォルトの編集ページで、必要に応じて次のいずれかを変更します。
 - a. クエリ結果で、Amazon S3 の結果の送信先、結果形式、または結果ファイルを変更します。
 - b. (オプション) サービスアクセスで、S3 送信先に最大 24 時間かかるクエリを配信する場合は、サービスロールの追加を選択して、チェックボックスをオンにするまでに最大 24 時間かかるクエリをサポートします。

完了までに最大 24 時間かかる大規模なクエリは、S3 送信先に配信されます。

チェックボックスをオンにしない場合、12 時間以内に完了したクエリのみが S3 ロケーションに配信されます。

- [新しいサービスロールを作成して使用] または [既存のサービスロールを使用] を選択して、[サービスアクセス] 許可を指定します。

Create and use a new service role

- AWS Clean Rooms は、このテーブルに必要なポリシーを持つサービスロールを作成します。
- デフォルトの [サービスロール名] は `cleanrooms-query-receiver-
<timestamp>` です。
- ロールを作成してポリシーをアタッチするアクセス許可が必要です。

Use an existing service role

1. ドロップダウンリストから [既存のサービスロール名] を選択します。

ロールを一覧表示するアクセス許可がある場合は、ロールのリストが表示されます。

ロールを一覧表示するアクセス許可がない場合は、使用するロールの Amazon リソースネーム (ARN) を入力できます。

2. [IAM で表示] 外部リンクを選択してサービスロールを表示します。

既存のサービスロールがない場合、[既存のサービスロールを使用] オプションは使用できません。

デフォルトでは、AWS Clean Rooms は既存のロールポリシーを更新して必要なアクセス許可を追加しようとしません。

Note

- AWS Clean Rooms には、分析ルールに従ってクエリを実行するためのアクセス許可が必要です。のアクセス許可の詳細については AWS Clean Rooms、「」を参照してください[AWS の 管理ポリシー AWS Clean Rooms](#)。
- ロールに十分なアクセス許可がない場合 AWS Clean Rooms、ロールに十分なアクセス許可がないことを示すエラーメッセージが表示されます AWS Clean Rooms。続行する前に、ロールポリシーを追加する必要があります。

- ロールポリシーを変更できない場合は、[がサービスロールのポリシーを見つけ AWS Clean Rooms](#) られなかったことを示すエラーメッセージが表示されます。

6. [Save changes] (変更の保存) をクリックします。
7. 更新した [クエリ結果設定] がコラボレーションの詳細ページに表示されます。

ジョブ結果設定のデフォルト値の編集

Note

Amazon S3 の結果の送信先をデータソースと同じ S3 バケット内にすることはできません。

結果を受け取ることができるメンバーは、AWS Clean Rooms コンソールでジョブ結果設定のデフォルト値を編集できます。

ジョブ結果設定のデフォルト値を編集するには

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [自身のメンバー能力] のステータスが [結果を受け取る] になっているコラボレーションを選択します。
4. 分析タブの結果設定のデフォルト で、編集 を選択します。
5. 結果設定のデフォルトの編集ページで、必要に応じて次のいずれかを変更します。
 - a. ジョブの結果で、Amazon S3 の結果の宛先を変更します。
 - b. サービスアクセスで、既存のサービスロール名を変更します。
6. [Save changes] (変更の保存) をクリックします。
7. 更新されたジョブ結果設定がコラボレーションの詳細ページに表示されます。

他の のクエリ出力の使用 AWS のサービス

SQL クエリ出力は、Clean Rooms ML モデルのシードデータに使用できます。詳細については、[「AWS Clean Rooms ML」](#)を参照してください。

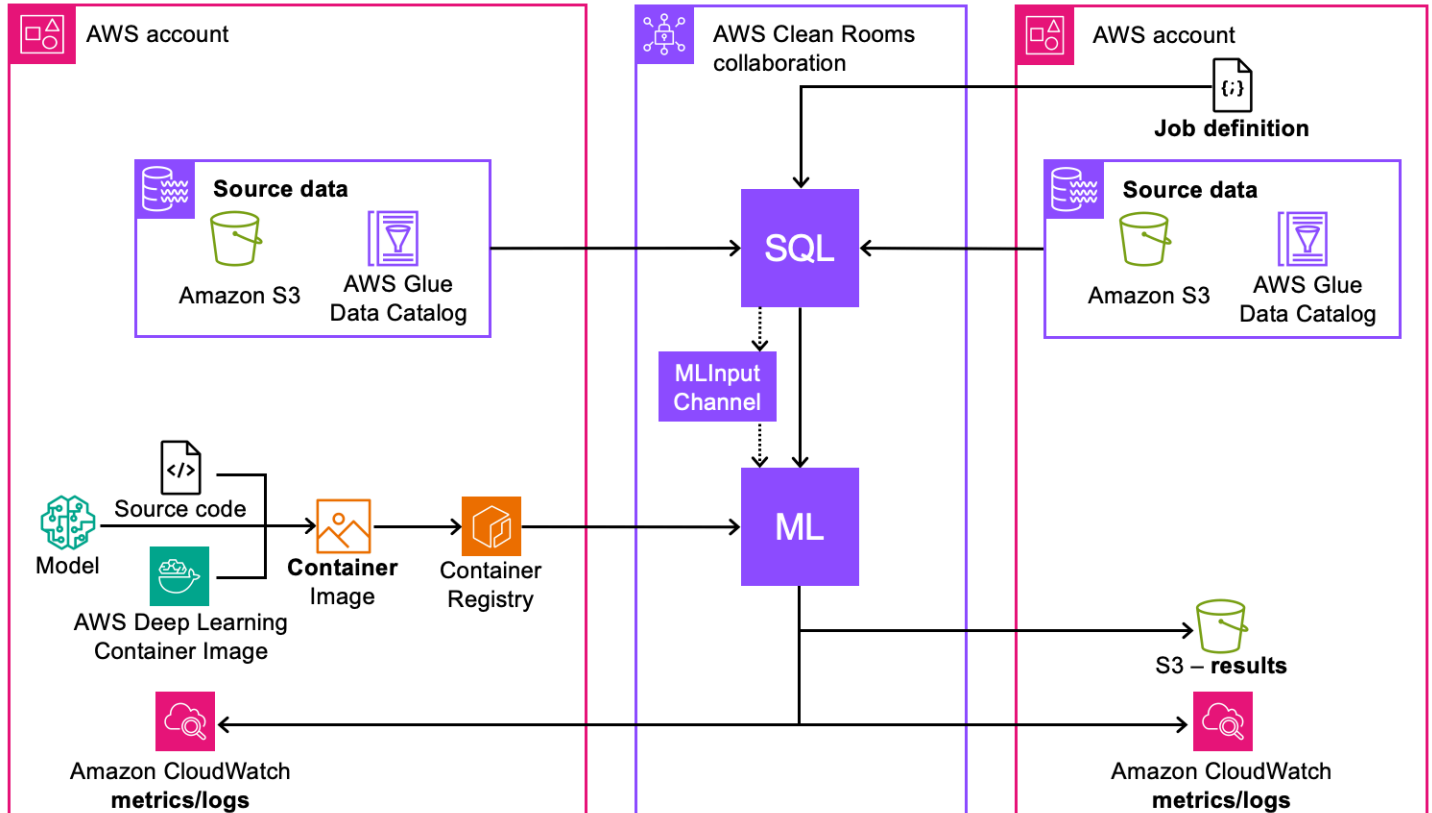
からのクエリ出力 AWS Clean Rooms は、コンソール (コンソールを使用してクエリを実行する場合) で使用でき、指定された Amazon S3 バケットにダウンロードされます。そこから、Amazon Quick や Amazon SageMaker AI AWS のサービスなどの他の のクエリ出力を、それらのサービスが Amazon S3 のデータをどのように使用するかに応じて使用できます。

Amazon Quick の詳細については、[「Amazon Quick Documentation」](#)を参照してください。

Amazon SageMaker AI の詳細については、[Amazon SageMakerドキュメント](#)」を参照してください。

AWS Clean Rooms ML でのカスタムモデリング

技術的な観点から、次の図は AWS Clean Rooms ML でのカスタム ML モデリングの仕組みを示しています。



Clean Rooms ML でのカスタム ML モデリングの仕組みは次のとおりです。

1. データソース設定

- ソースデータは、Amazon S3 カタログ、AWS Glue Data Catalog、または Snowflake に保存できます。
- AWS Glue Data Catalog は、整理とカタログ化に使用されます。
- 複数の からのデータは AWS アカウント、同じコラボレーション内で使用できます。

2. SQL クエリとデータ処理

- SQL クエリはソースデータへのアクセスと処理に使用されます
- クエリは AWS Clean Rooms コラボレーションの境界内で実行されます。
- モデルトレーニングのために ML 入力チャンネルに処理されたデータフィード

3. ML モデル開発

- モデルのソースコードは、AWS 深層学習コンテナイメージを使用して開発できます。
- カスタムコンテナイメージを作成して Amazon Elastic Container Registry に保存する必要があります

4. インフラストラクチャコンポーネント

- Amazon Elastic Container Registry は ML モデルコンテナを保存および管理します
- ML 処理は安全な AWS Clean Rooms コラボレーション環境内で行われます

5. モニタリングとロギング

- Amazon CloudWatch は、共同作業者の両方のメトリクスとログを提供します。
- モニタリングは、コラボレーション AWS アカウント に関与している 全体で利用できます。
- パフォーマンスメトリクスと運用ログに関係者がアクセスできる

6. 結果管理

- 結果へのアクセスは、コラボレーションのアクセス許可に従って制御されます。

開始する前に、[カスタム ML モデリングの前提条件](#)「」および[トレーニングコンテナのモデル作成ガイドライン](#)「」を参照してください。

プライバシー強化合成データセットの生成

合成データセットには、基になる元のデータセットと同様の統計プロパティがありますが、元のデータセットに存在する実際の観測値は含まれません。プライバシーが強化された合成データセットを使用することで、データプライバシーの懸念が以前に妨げられていた新しい機械学習 (ML) モデルトレーニングのユースケースをロック解除できます。ML 入力チャネルを作成すると、ML モデルのトレーニング中に機密情報を保護するための合成データを生成できます。

合成データを使用してテンプレートを作成するときは、次のことを行う必要があります。

- テンプレート出力を合成する必要がある
- 出力スキーマ列を数値またはカテゴリとして分類する
- 組織のニーズに基づいて合成データをカスタマイズする
- プライバシー設定を調整します。
 - プライバシーレベルを設定する (イプシロン)
 - プライバシーしきい値を設定する

⚠ Warning

合成データ生成は、特定の個人が元のデータセットに存在するか、それらの個人の学習属性が存在するかにかかわらず、個々の属性の推測から保護します。ただし、個人を特定できる情報 (PII) を含む元のデータセットのリテラル値が合成データセットに表示されるのを防ぐことはできません。

1 つのデータサブジェクトのみに関連付けられている入力データセット内の値は、データセットを再識別する可能性があるため、避けることをお勧めします。たとえば、郵便番号にユーザーが 1 人しかいない場合、合成データセットにその郵便番号が存在すると、そのユーザーが元のデータセットに属していたことが確認されます。高精度値の切り捨てや、まれなカタログを他のカタログに置き換えるなどの手法を使用して、このリスクを軽減できます。これらの変換は、ML 入力チャネルの作成に使用されるクエリの一部にすることができます。

カスタムモデルトレーニングの合成データを生成する方法の詳細については、「」を参照してください [SQL 分析テンプレートの作成](#)。

合成出力を含む分析テンプレートは、ML 入力チャネルの作成にのみ使用できます。詳細については、「[AWS Clean Rooms ML での ML 入力チャネルの作成](#)」を参照してください。

合成データ生成に関する考慮事項

AWS Clean Rooms ML を使用すると、コラボレーションメンバーは、元のデータセットのサブジェクトを集合データセットから不可逆的に識別解除してカスタム機械学習モデルをトレーニングする合成データセットを作成できます。コラボレーションを作成するときは、合成データ生成に対して誰が支払うかを指定するように支払い情報を設定する必要があります。合成データセットを生成し、カスタム機械学習モデルをトレーニングするための大まかな手順は次のとおりです。

1. コラボレーションメンバーは、以下を含む分析テンプレートを作成します。
 - 合成するデータセットを定義するために必要な SQL。
 - 合成データがデータプロバイダーのコンプライアンス要件を満たしていることを確認するために使用されるプライバシー関連の設定。
2. すべてのデータプロバイダーが分析テンプレートを承認すると、コラボレーションクエリランナーはテンプレートを使用して機械学習 (ML) 入力チャネルを作成します。
3. Clean Rooms ML は合成データセットを生成し、分析テンプレートで指定されたプライバシーしきい値を満たしていることを確認します。

4. すべてのしきい値が満たされると、ML 入力チャネルに合成データセットが入力されます。
5. その後、この ML 入力チャネルを使用して、コラボレーションに関連付けられたカスタム ML モデルをトレーニングできます。

重要な考慮事項:

- Clean Rooms ML で生成された合成データは、元のデータセットで見つかった個人を特定できる情報 (PII) を含む個々の値を削除、編集、難読化、またはサニタイズしません。合成データセットは、元のデータセットからレコード全体ではなく値をサンプリングすることによって生成されます。
- 元のデータセットに同様の行が含まれている場合、合成データには元のデータセットの行と同じように見える行が含まれている可能性があります。

データセットの準備:

- クラス分散が著しく不均衡な列は避けてください。これは、予測値または「Y」列にとって特に重要です。極端な不均衡により、合成データセットの全体的なプライバシーが低下します。
- Clean Rooms ML は、シーケンシャルレコード間の相関関係を維持することが重要な時系列データからの合成データの生成をサポートしていません。
- Clean Rooms ML は、テキストデータまたは非構造化データからの合成データの生成をサポートしていません。
- 以下のデータ型 (タイプ) がサポートされています。

デー
タ
型
名

BIGINT

BOOLEAN

CHAR

DATE

デー
タ
型
名

DECIMAL

FLOAT

INTEGER

LONG

REAL

SHORT

SMALLINT

TIME

TIMESTAMP
_LTZ

TIMESTAMP
_NTZ

TINYINT

VARCHAR

機能制限:

- 合成データ生成の場合、予測列の最大数は 1 です。
- ターゲット列がカテゴリ別である場合、元のデータセットのカテゴリの最大数は 100 です。
- 元のデータセットでは、行数は 1,500 ~ 250 万で、最大列数は 1,000 です。ターゲット列の null 以外の値の場合、最小行数は 1,000 です。

プライバシーメトリクス:

- Clean Rooms ML は、生成された合成データがメンバーシップ推論攻撃 (MIAs) に対してどのように保護されているかを測定するプライバシースコアを提供します。このサービスは、合成プロセスから元のデータの 5% を保持して、このスコアを計算します。
- 50% に近いスコアは良好と見なされます。スコアが高いほど、MIAs に対する保護が低いことを示します。スコアが 50% を大幅に下回ることはまれであり、合成されたデータ内の元のデータからパターンが表現されないことが原因である可能性があります。

ダウンストリームカスタムモデル:

- Clean Rooms ML で生成された合成データは、最大 5 つのクラスを持つ二項分類モデルと複数クラス分類モデルのトレーニングに最適です。
- Clean Rooms ML で生成された合成データを使用して回帰モデルをトレーニングすると、二乗平均平方根誤差 (RMSE) で測定されるモデルの精度が低下する可能性があります。

AWS Clean Rooms ML でのコラボレーションの作成と参加

コラボレーションクリエイターは、コラボレーションの作成、メンバーの招待、ロールの割り当てを担当します。招待されたメンバーはコラボレーションに参加し、コラボレーションの設定方法に応じて、結果設定、トレーニング済みモデルアーティファクトの送信先設定を指定し、支払い責任を受け入れます。

機械学習用のコラボレーションの作成

次の手順は、機械学習用のコラボレーションの作成、1 人以上のメンバーの招待、モデルトレーニングの開始、結果の受信、モデルアーティファクトやメトリクスを含むトレーニング済みモデル結果の受信、モデル推論結果の受信が可能なメンバーの割り当てを行う方法を示しています。コラボレーションクリエイターは、クエリコンピューティング、モデルトレーニング、モデル推論のコストを支払うメンバーも割り当てます。

Console

機械学習のコラボレーションを作成するには (コンソール)

1. [コラボレーションを作成し、1 人以上のメンバーをコラボレーションに招待する](#)
2. クエリとジョブを使用して、分析に次のメンバー機能を割り当てます。
 - モデルトレーニングを開始するメンバーに実行クエリを割り当てます。

- クエリ結果を受け取るメンバーに、分析からの受信結果を割り当てる。
3. 専用ワークフローを使用して、ML モデリングに次のメンバー機能を割り当てます。
 - モデルアーティファクトやメトリクスなど、トレーニング済みモデルの結果を受け取るメンバーに、トレーニング済みモデルからの受信出力を割り当てます。
 - モデル推論の結果を受け取るメンバーにモデル推論からの受信出力を割り当てます。
 4. 支払いを設定するには、クエリコンピューティング、モデルトレーニング、モデル推論、合成データセット生成のコストを支払うメンバーを指定します。

合成データセットには、基になる元のデータセットと同様の統計プロパティがありますが、元のデータセットに存在する実際の観測値は含まれません。プライバシーが強化された合成データセットを使用することで、データプライバシーの懸念が以前に妨げられていた新しい ML モデルトレーニングのユースケースをロック解除できます。詳細については、[「プライバシー強化合成データ生成」](#)を参照してください。

これらのコストはそれぞれ、同じメンバーまたは異なるメンバーに割り当てることができます。招待されたメンバーが支払い費用の支払いを担当するメンバーである場合は、コラボレーションに参加する前に支払い責任を引き受ける必要があります。

5. メンバーシップを設定する場合、コラボレーション作成者は今すぐメンバーシップに参加するか、後でメンバーシップを作成するかを決定できます。次に、コラボレーションクリエーターは ML 設定をセットアップする必要があります。
 - a. コラボレーション作成者が結果レシーバーでもある場合は、結果設定のデフォルトにクエリ結果の送信先と形式も指定する必要があります。
 - b. ML 設定は、Clean Rooms ML がメトリクスを発行するためのロールを提供します AWS アカウント。コラボレーション作成者がトレーニング済みモデルアーティファクトも受信している場合は、結果の受信に使用する Amazon S3 バケットを指定できます。
 - c. ML 設定セクションで、ML 設定の作成を選択し、Amazon S3 のモデル出力先と、この場所へのアクセスに必要なサービスアクセスロールを指定します。
 - d. コラボレーション作成者が支払いコストの支払いを担当するメンバーである場合は、コラボレーションを作成する前に支払い責任を引き受ける必要があります。

API

機械学習 (API) のコラボレーションを作成するには

1. [コラボレーションを作成し、1人以上のメンバーをコラボレーションに招待する](#)
2. コラボレーションメンバーに次のロールを割り当てます。
 - CAN_QUERY - モデルトレーニングと推論を開始するメンバーに割り当てられます。
 - CAN_RECEIVE_MODEL_OUTPUT - トレーニング済みモデル結果を受け取るメンバーに割り当てられます。
 - CAN_RECEIVE_INFERENCE_OUTPUT - モデル推論結果を受け取るメンバーに割り当てられます。

コラボレーション作成者が結果レシーバーでもある場合は、コラボレーションの作成時にクエリ結果の送信先と形式も指定する必要があります。また、クエリ結果の送信先に結果を書き込むサービスロール Amazon リソースネーム (ARN) も提供します。

3. クエリコンピューティング、モデルトレーニング、モデル推論のコストを支払うメンバーを指定します。これらのコストはそれぞれ、同じメンバーまたは異なるメンバーに割り当てることができます。招待されたメンバーが支払い費用の支払いを担当するメンバーである場合は、コラボレーションに参加する前に支払い責任を引き受ける必要があります。
4. 次のコードはコラボレーションを作成し、クエリを実行して結果を受け取ることができるメンバーを招待し、コラボレーション作成者をモデルアーティファクトレシーバーとして指定します。

```
import boto3
acr_client= boto3.client('cleanrooms')

collaboration = a_acr_client.create_collaboration(
    members=[
        {
            'accountId': 'invited_member_accountId',
            'memberAbilities': ["CAN_QUERY", "CAN_RECEIVE_RESULTS"],
            'displayName': 'member_display_name'
        }
    ],
    name='collaboration_name',
    description=collaboration_description,
    creatorMLMemberAbilities= {
        'customMLMemberAbilities': ["CAN_RECEIVE_MODEL_OUTPUT",
"CAN_RECEIVE_INFERENCE_OUTPUT"],
    },
    creatorDisplayName='creator_display_name',
    queryLogStatus="ENABLED",
    analyticsEngine="SPARK",
```

```

        creatorPaymentConfiguration={
            "queryCompute": {
                "isResponsible": True
            },
            "machineLearning": {
                "modelTraining": {
                    "isResponsible": True
                },
                "modelInference": {
                    "isResponsible": True
                }
            }
        }
    )

    collaboration_id = collaboration['collaboration']['id']
    print("collaborationId: {collaboration_id}")

    member_membership = a_acr_client.create_membership(
        collaborationIdentifier = collaboration_id,
        queryLogStatus = 'ENABLED',
        paymentConfiguration={
            "queryCompute": {
                "isResponsible": True
            },
            "machineLearning": {
                "modelTraining": {
                    "isResponsible": True
                },
                "modelInference": {
                    "isResponsible": True
                }
            }
        }
    )

```

5. 次に、コラボレーションクリエイターは ML 設定をセットアップする必要があります。ML 設定は、Clean Rooms ML がメトリクスとログを に発行するためのロールを提供します AWS アカウント。コラボレーション作成者が結果 (モデルアーティファクトまたは推論結果) も受信している場合は、結果の受信に使用する Amazon S3 バケットを指定できます。

```

import boto3
acr_ml_client= boto3.client('cleanroomsm1')

```

```
acr_ml_client.put_ml_configuration(  
    membershipIdentifier=membership_id,  
    defaultOutputLocation={  
        'roleArn': 'arn:aws:iam::account:role/roleName',  
        'destination': {  
            's3Destination': {  
                's3Uri': "s3://bucketName/prefix"  
            }  
        }  
    }  
)
```

コラボレーションへの参加

コラボレーション作成者がタスクを完了したら、招待されたメンバーはタスクを完了する必要があります。

Console

メンバーシップを作成してコラボレーションに参加するには (コンソール)

1. 招待されたメンバーは [メンバーシップを作成し、コラボレーションに参加します](#)。
2. 招待されたメンバーが、クエリコンピューティング、モデルトレーニング、モデル推論コストなどの支払いを担当するメンバーである場合は、コラボレーションに参加する前に支払い責任を引き受ける必要があります。
3. 招待されたメンバーは ML 設定をセットアップします。これにより、Clean Rooms ML がモデルメトリクスを発行するロールが提供されます AWS アカウント。トレーニング済みモデルアーティファクトを受け取るメンバーでもある場合は、トレーニング済みモデルアーティファクトが保存されている Amazon S3 バケットを提供する必要があります。

API

メンバーシップを作成してコラボレーションに参加するには (API)

1. 招待されたメンバーが結果を受け取れるメンバーである場合は、そのメンバーがクエリ結果の送信先と形式を指定します。また、サービスがクエリ結果の送信先に書き込むことを許可するサービスロール ARN も提供します。

招待されたメンバーが、クエリコンピューティング、モデルトレーニング、モデル推論コストなどの支払いを担当するメンバーである場合は、コラボレーションに参加する前に支払い責任を引き受ける必要があります。

招待されたメンバーがカスタムモデリングのモデルトレーニングとモデル推論の支払いを担当するメンバーである場合は、コラボレーションに参加する前に支払い責任を引き受ける必要があります。

次のコードは、クエリログ記録を有効にしてメンバーシップを作成します。

```
import boto3
acr_client= boto3.client('cleanrooms')

acr_client.create_membership(
    membershipIdentifier='membership_id',
    queryLogStatus='ENABLED'
)
```

2. 招待されたメンバーは ML 設定をセットアップします。これにより、Clean Rooms ML がモデルメトリクスを に発行するロールが提供されます AWS アカウント。トレーニング済みモデルアーティファクトを受け取るメンバーでもある場合は、トレーニング済みモデルアーティファクトが保存されている Amazon S3 バケットを提供する必要があります。

```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')

acr_ml_client.put_ml_configuration(
    membershipIdentifier='membership_id',
    defaultOutputLocation={
        'roleArn': "arn:aws:iam::account:role/role_name",
        'destination': {
            's3Destination': {
                's3Uri': "s3://bucket_name/prefix"
            }
        }
    }
)
```

AWS Clean Rooms ML でのトレーニングデータの提供

コラボレーション作成者がコラボレーションを作成し、招待されたメンバーが参加したら、トレーニングデータをコラボレーションに提供する準備が整います。どのメンバーもトレーニングデータを提供できます。

Console

トレーニングデータを提供するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. Tables ページで、Configure new table を選択します。
4. 新しいテーブルを設定する データソース で、Amazon S3、Amazon Athena、または Snowflake を選択し、データソースに基づいて次の手順を実行します。

を使用している場合	THEN
Amazon S3	1. ドロップダウンリストからデータベースを選択し、データベースからテーブルを選択します。 2. [コラボレーションで許可された列] で、[すべての列] または [カスタムリスト] を選択します。 3. 設定済みテーブルの詳細については、このテーブルの名前とオプションの説明を指定します。 4. モデルメトリクスを報告する場合は、メトリクスの名前と、出力ログを検索してメトリクスを検索する正規表現ステートメントを入力します。 5. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。

を使用している場合	THEN
Amazon Athena	1. ドロップダウンリストからデータベースを選択し、データベースからテーブルを選択します。 2. [コラボレーションで許可された列] で、[すべての列] または [カスタムリスト] を選択します。 3. 設定済みテーブルの詳細については、このテーブルの名前とオプションの説明を指定します。 4. モデルメトリクスをレポートする場合は、メトリクスの名前と、出力ログを検索してメトリクスを検索する正規表現ステートメントを入力します。 5. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
Snowflake	1. 既存のシークレット ARN を使用するか、このテーブルの新しいシークレットを保存して、Snowflake 認証情報を指定します。 2. Snowflake テーブルとスキーマの詳細には、詳細を手動で入力するか、詳細を自動的にインポートします。 3. スキーマには、列名を入力し、ドロップダウンリストからデータ型を選択します。 4. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。

5. [新しいテーブルを設定] を選択します。
6. テーブルの詳細ページで、分析ルールを設定を選択して、このテーブルのカスタム分析ルールを設定します。カスタム分析ルールは、データへのアクセスを制限します。データに対し

て事前に承認された特定のクエリセットを許可するか、特定のアカウントのセットにデータのクエリを許可することができます。

- a. 分析ルールタイプでカスタムを選択し、作成方法でガイド付きフローを選択します。
 - b. [次へ] を選択します。
 - c. 分析コントロールを指定する では、新しい各分析を確認する と、特定の共同作業者による分析を許可する のどちらかを選択します。
 - d. [次へ] を選択します。
 - e. (オプション) 分析結果のコントロールを指定する では、出力で許可されていない列に、出力から列を除外するかどうかを指定します。None を選択した場合、出力から除外される列はありません。カスタムリストを選択した場合は、出力から削除する特定の列を指定できます。
 - f. 出力に適用される追加分析では、結果を生成する前に追加の分析を許可、拒否、または要求するかどうかを指定します。
 - g. [次へ] を選択します。
 - h. (オプション) 差分プライバシーを設定する で、無効化を選択します。
 - i. [次へ] を選択します。
 - j. 「確認と設定」ページの情報を確認し、「分析ルールの設定」を選択します。
7. テーブルの詳細ページで、コラボレーションに関連付けるを選択します。
 8. テーブルの関連付けダイアログボックスで、このテーブルに関連付けるコラボレーションを選択し、コラボレーションの選択を選択します。
 9. テーブルの関連付けページで、テーブルの関連付けの詳細、サービスアクセス、タグの情報を確認して確認します。[テーブルに関連付ける] を選択します。
 10. テーブルに関連付けられたテーブルで、先ほど関連付けたテーブルの横にあるラジオボタンを選択します。Actions メニューから、コラボレーション分析ルールグループの Configure を選択します。
 11. コラボレーション分析ルールの設定ページで、許可される追加分析について、コラボレーションメンバーまたは特定のコラボレーションメンバーが追加分析を実行できるかどうかを選択します。

結果配信では、クエリ出力から結果を受信できるメンバーを選択します。

12. [分析ルールを設定] を選択します。

API

トレーニングデータを提供するには (API)

1. AWS Glue テーブルと使用できる列を指定 AWS Clean Rooms して、で使用する既存のテーブルを設定します。

特定のパラメータを使用して次のコードを実行します。

```
import boto3
acr_client= boto3.client('cleanrooms')

acr_client.create_configured_table(
    name='configured_table_name',
    tableReference= {
        'glue': {
            'tableName': 'glue_table_name',
            'databaseName': 'glue_database_name'
        }
    },
    analysisMethod="DIRECT_QUERY",
    allowedColumns=["column1", "column2", "column3",...]
)
```

2. データへのアクセスを制限するカスタム分析ルールを設定します。データに対して事前に承認された特定のクエリセットを許可するか、特定のアカウントのセットにデータのクエリを許可することができます。

特定のパラメータを使用して次のコードを実行します。

```
import boto3
acr_client= boto3.client('cleanrooms')

acr_client.create_configured_table_analysis_rule(
    configuredTableIdentifier='configured_table_id',
    analysisRuleType='CUSTOM',
    analysisRulePolicy= {
        'v1': {
            'custom': {
                'allowedAnalyses': ['ANY_QUERY'],
                'allowedAnalysisProviders': ['query_runner_account'],
                'additionalAnalyses': "REQUIRED"
            }
        }
    }
)
```



```
    }
  }
)
```

この例では、特定のアカウントがデータに対してクエリを実行でき、追加の分析が必要です。

3. 設定済みテーブルをコラボレーションに関連付け、サービスアクセスロールを AWS Glue テーブルに提供します。

特定のパラメータで次のコードを実行します。

```
import boto3
acr_client= boto3.client('cleanrooms')

acr_client.create_configured_table_association(
    name='configured_table_association_name',
    membershipIdentifier='membership_id',
    configuredTableIdentifier='configured_table_id',
    roleArn='arn:aws:iam::account:role/role_name'
)
```

Note

このサービスロールにはテーブルに対するアクセス許可があります。サービスロールは、ガクエリ AWS Clean Rooms を実行できるメンバーに代わって許可されたクエリを実行する場合にのみ引き受けることができます。コラボレーションメンバー (データの所有者以外) は誰も、コラボレーションの基になるテーブルにはアクセスできません。データ所有者は差分プライバシーをオフにして、テーブルを他のメンバーによるクエリに使用できるようにします。

4. 最後に、設定されたテーブルの関連付けに分析ルールを追加します。

特定のパラメータで次のコードを実行します。

```
import boto3
acr_client= boto3.client('cleanrooms')

acr_client.create_configured_table_association_analysis_rule(

    configuredTableAssociationIdentifier='configured_table_association_identifier',
    membershipIdentifier='membership_id',
```

```

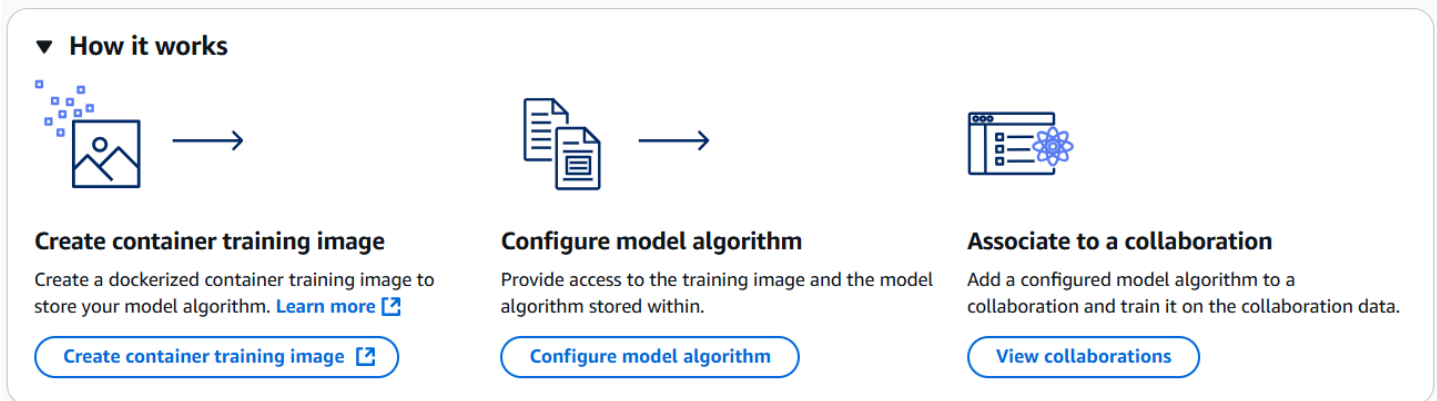
configuredTableIdentifier='configured_table_id',
analysisRuleType = 'CUSTOM',
analysisRulePolicy= {
  'v1': {
    'custom': {
      'allowedAdditionalAnalyses':
['configured_model_algorithm_association_arns'],
      'allowedResultReceivers': ['query_runner_account']
    }
  }
}
)

```

AWS Clean Rooms ML でのモデルアルゴリズムの設定

[コンテナトレーニングイメージを作成](#)したら、モデルアルゴリズムを設定する必要があります。モデルアルゴリズムを設定すると、コラボレーションに関連付けることができます。

次の図は、コンテナトレーニングイメージを作成した後、コラボレーションに関連付ける前に、モデルアルゴリズムをステップとして設定する方法を示しています。



Console

カスタム ML モデルアルゴリズムを設定するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左側のナビゲーションペインで、カスタム ML モデルを選択します。
3. カスタム ML モデルページで、モデルアルゴリズムの設定を選択します。

4. モデルアルゴリズムの設定ページで、モデルアルゴリズムの詳細に名前とオプションの説明を入力します。
5. モデルトレーニングを実行する場合は、トレーニングイメージ ECR コンテナの詳細について、
 - a. トレーニングイメージの URI を指定 チェックボックスをオンにします。
 - b. ドロップダウンリストから、トレーニングモデル、推論コンテナ、またはその両方を持つリポジトリを選択します。
 - c. イメージを選択します。
 - d. (オプション) トレーニングイメージにアクセスするためのエントリポイントの値を入力します。
 - e. (オプション) 引数の値を入力します。
6. (オプション) モデルメトリクスをレポートする場合は、トレーニングメトリクスに、メトリクスの名前と、出力ログを検索してメトリクスを検索する正規表現ステートメントを入力します。
7. モデル推論を実行する場合は、推論イメージ ECR コンテナの詳細について、
 - a. 推論イメージの URI を指定 チェックボックスをオンにします。
 - b. ドロップダウンリストからリポジトリを選択します。
 - c. イメージを選択します。
8. [サービスアクセス] では、このテーブルへのアクセスに使用する [既存のサービスロール名] を選択します。
9. 暗号化 で、暗号化設定をカスタマイズ を選択して、独自の KMS キーと関連情報を指定します。それ以外の場合、Clean Rooms ML が暗号化を管理します。
10. タグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
11. モデルアルゴリズムの設定 を選択します。

API

カスタム ML モデルアルゴリズム (API) を設定するには

1. SageMaker AI 互換ドッカーイメージを作成します。Clean Rooms ML は、SageMaker AI 互換のドッカーイメージのみをサポートします。

2. SageMaker AI 互換のドッカーイメージを作成したら、Amazon ECR を使用してトレーニングイメージを作成します。[Amazon Elastic Container Registry ユーザーガイド](#)の指示に従って、コンテナトレーニングイメージを作成します。
3. Clean Rooms ML で使用するモデルアルゴリズムを設定します。次の情報を提供する必要があります。
 - Amazon ECR リポジトリリンクと、モデルをトレーニングして推論を実行するための追加の引数。Clean Rooms ML は、推論コンテナでのバッチ変換ジョブの実行をサポートしています。
 - Clean Rooms ML がリポジトリにアクセスできるようにするサービスアクセスロール。
 - (オプション) 推論コンテナ。これは別の設定済みモデルアルゴリズムで指定できますが、トレーニングコンテナと推論コンテナの両方が同じリソースの一部として管理されるように、このステップで指定することをお勧めします。

特定のパラメータで次のコードを実行します。

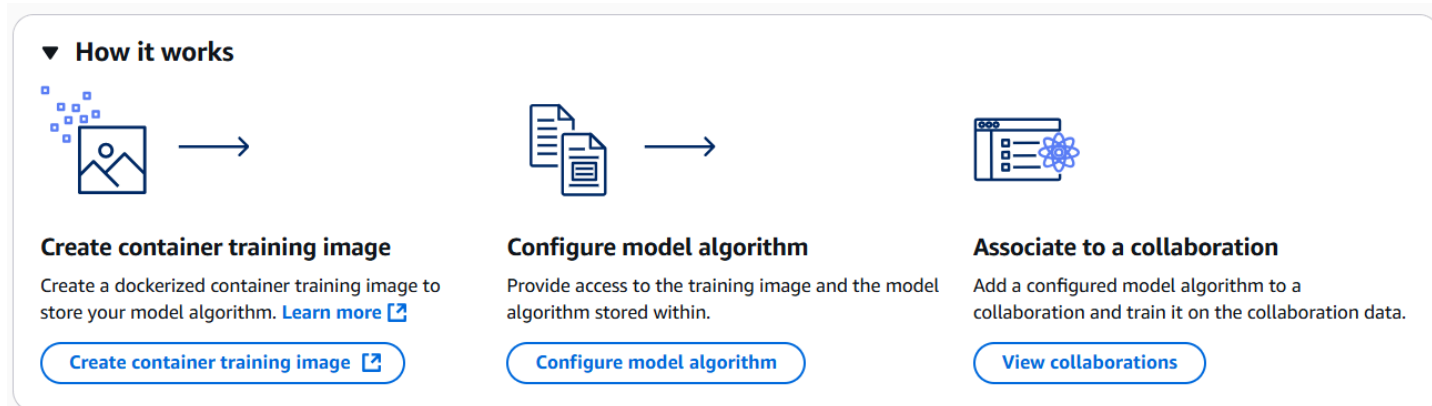
```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')

acr_ml_client.create_configured_model_algorithm(
    name='configured_model_algorithm_name',
    trainingContainerConfig={
        'imageUri': 'account.dkr.ecr.region.amazonaws.com/image_name:tag',
        'metricDefinitions': [
            {
                'name': 'custom_metric_name_1',
                'regex': 'custom_metric_regex_1'
            }
        ]
    },
    inferenceContainerConfig={
        'imageUri': 'account.dkr.ecr.region.amazonaws.com/image_name:tag',
    },
    roleArn='arn:aws:iam::account:role/role_name'
)
```

AWS Clean Rooms ML で設定されたモデルアルゴリズムの関連付け

モデルアルゴリズムを設定したら、モデルアルゴリズムをコラボレーションに関連付ける準備が整います。モデルアルゴリズムを関連付けると、モデルアルゴリズムはコラボレーションのすべてのメンバーが使用できるようになります。

次の図は、コンテナトレーニングイメージを作成し、モデルアルゴリズムを設定した後、設定されたモデルアルゴリズムを最後のステップとして関連付ける方法を示しています。



Console

Note

モデルアルゴリズムを関連付けた後は、編集できません。変更するには、関連付けられたモデルアルゴリズムを削除し、新しいモデルアルゴリズムを関連付けます。

カスタム ML モデルアルゴリズムを関連付けるには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左側のナビゲーションペインで、カスタム ML モデルを選択します。
3. カスタム ML モデルページで、コラボレーションに関連付ける設定済みモデルアルゴリズムを選択し、コラボレーションに関連付けるを選択します。
4. 設定されたモデルアルゴリズムの関連付けウィンドウで、関連付けるコラボレーションを選択します。
5. [コラボレーションを選ぶ] を選択します。

6. モデルアルゴリズムの関連付けページで、モデルアルゴリズムの関連付けの詳細に名前とオプションの説明を入力します。
7. Model algorithm で、Configured model algorithm を選択します。
8. トレーニング済みモデルエクスポートのプライバシー設定の場合、
 - a. モデルファイルをエクスポートするには、モデルファイルのチェックボックスをオンにします。
 - b. 出力ファイルをエクスポートするには、出力ファイルのチェックボックスをオンにします。
 - c. エクスポートされたデータの最大サイズ値を入力します。値は 0.01 ~ 10 の間でなければなりません。
9. (オプション) トレーニング済みモデル推論ジョブのプライバシー設定で、完全なエラーログまたは短いエラー概要をメンバーに送信する場合は、
 - a. フルログで、ドロップダウンリストから 1 つ以上のアカウント IDs を選択します。
 - b. (オプション) フィルターパターンに一致するログを送信する場合は、フィルターパターンを入力します。
 - c. (オプション) 別のアカウントとオプションのフィルターパターンを追加する場合は、ログポリシーの追加を選択します。
 - d. エラーの概要で、ドロップダウンリストから 1 つ以上のアカウント IDs を選択します。
 - e. (オプション) 編集する 1 つ以上のエンティティを選択して、エラーログまたはエラー概要から編集するエンティティを指定します。
 - PII – 個人を特定できる情報の編集
 - 数値 – 数値の編集
 - カスタム – カスタム秘匿化パターンに基づいて秘匿化
 - i. 前のステップでカスタム を選択した場合は、カスタムリダクションパターンを入力します。これにより、このパターンに一致する情報がログに記録されます。
 - ii. (オプション) 別のカスタムリダクションパターンを追加する場合は、別のカスタムパターンを追加するを選択します。
10. (オプション) トレーニング済みモデルメトリクスを設定する場合は、トレーニング済みモデルメトリクス設定で、ドロップダウンリストからノイズレベルを選択します。

None、Low、Medium、High を選択できます。

11. (オプション) アーティファクトの最大サイズを設定する場合は、アーティファクトの最大サイズ値を入力します。値は 0.01 ~ 10 の間でなければなりません。
12. (オプション) タグを有効にする場合は、新しいタグを追加を選択し、キーと値のペアを入力します。
13. 関連付ける を選択してください。

API

カスタム ML モデルアルゴリズムを関連付けるには (API)

特定のパラメータで次のコードを実行します。

また、異なるログにアクセスできるユーザーを定義し、顧客が正規表現を定義できるようにし、トレーニングモデルの出力または推論結果からエクスポートできるデータの量を定義するプライバシーポリシーも提供します。

Note

設定されたモデルアルゴリズムの関連付けはイミュータブルです。

```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')

acr_ml_client.create_configured_model_algorithm_association(
    name='configured_model_algorithm_association_name',
    description='purpose of the association',
    configuredModelAlgorithmArn='arn:aws:cleanrooms-ml:region:account:membership/
membershipIdentifier/configured-model-algorithm/identifier',
    privacyConfiguration={
        "policies": {
            "trainedModelExports": {
                "filesToExport": ['files to export'],
                "containerLogs": [
                    {
                        "allowedAccountIds": ['member_account_id'],
                        "filterPattern": ['filter pattern'],
                        "logRedactionConfiguration": {
                            "entitiesToRedact": [
                                'ALL_PERSONALLY_IDENTIFIABLE_INFORMATION',
                                'NUMBERS',
```

```

        'CUSTOM'
    ],
    "customEntityConfig": {
        "customDataIdentifiers": [
            'custom_regex_1',
            'custom_regex_2'
        ]
    }
}

},
"containerMetrics": {
    "noiseLevel": 'noise value'
},
"maxArtifactSize": {
    "unit": 'unit',
    "value": 'number'
}
},
"trainedModelInferenceJobs": {
    "containerLogs": [
        {
            "allowedAccountIds": ['member_account_id'],
            "filterPattern": ['filter pattern'],
            "logRedactionConfiguration": {
                "entitiesToRedact": [
                    'ALL_PERSONALLY_IDENTIFIABLE_INFORMATION',
                    'NUMBERS',
                    'CUSTOM'
                ],
                "customEntityConfig": {
                    "customDataIdentifiers": [
                        'custom_regex_1',
                        'custom_regex_2'
                    ]
                }
            }
        ]
    },
    "maxOutputSize": {
        "unit": 'unit',
        "value": 'number'
    }
}
}

```



```

    }
  },
  tags={
    'tag': 'tag'
  }
)

```

設定されたモデルアルゴリズムがコラボレーションに関連付けられると、トレーニングデータプロバイダーはテーブルにコラボレーション分析ルールを追加する必要があります。このルールにより、設定されたモデルアルゴリズムの関連付けが設定されたテーブルにアクセスできるようになります。すべての貢献するトレーニングデータプロバイダーは、次のコードを実行する必要があります。

```

import boto3
acr_client= boto3.client('cleanrooms')

acr_client.create_configured_table_association_analysis_rule(
    membershipIdentifier= 'membership_id',
    configuredTableAssociationIdentifier= 'configured_table_association_id',
    analysisRuleType= 'CUSTOM',
    analysisRulePolicy = {
        'v1': {
            'custom': {
                'allowedAdditionalAnalyses': ['arn:aws:cleanrooms-
ml:region:*:membership/*/configured-model-algorithm-association/*'],
                'allowedResultReceivers': []
            }
        }
    }
)

```

Note

設定されたモデルアルゴリズムの関連付けはイミュータブルであるため、カスタムモデル設定の最初の数回の反復allowedAdditionalAnalysesでワイルドカードを使用できるようにモデルを一覧表示したいデータプロバイダーをトレーニングすることをお勧めします。これにより、モデルプロバイダーは、更新されたモデルコードをデータでトレーニングする前に、他のトレーニングプロバイダーが再関連付けする必要なく、コードを繰り返し実行できます。

AWS Clean Rooms ML での ML 入力チャネルの作成

前提条件:

- にアクセスできる AWS アカウント 。 AWS Clean Rooms
- ML 入力チャネルを作成する AWS Clean Rooms にセットアップされたコラボレーション
- コラボレーションでデータをクエリし、ML 入力チャネルを作成するアクセス許可。
- (オプション) ML 入力チャネルに関連付ける既存のモデルアルゴリズム、または新しい入力チャネルを作成するアクセス許可
- (オプション) 指定したモデルに対して実行できる分析ルールを含むテーブル。
- (オプション) データセットの生成に使用する既存の SQL クエリまたは分析テンプレート
- (オプション) 適切なアクセス許可、または新しいサービスロールを作成するためのアクセス許可を持つ既存のサービスロール
- (オプション) 独自の暗号化 AWS KMS キーを使用する場合のカスタムキー
- コラボレーションで ML モデルを作成および管理するための適切なアクセス許可

ML 入力チャネルは、特定のデータクエリから作成されるデータセットです。データをクエリできるメンバーは、ML 入力チャネルを作成することで、トレーニングと推論のためにデータを準備できます。ML 入力チャネルを作成すると、そのデータを同じコラボレーション内のさまざまなトレーニングモデルで使用できます。トレーニングと推論用に個別の ML 入力チャネルを作成する必要があります。

ML 入力チャネルを作成するには、入力データのクエリと ML 入力チャネルの作成に使用される SQL クエリを指定する必要があります。このクエリの結果はメンバーと共有されることはなく、Clean Rooms ML の境界内にとどまります。リファレンス Amazon リソースネーム (ARN) は、次のステップでモデルのトレーニングや推論の実行に使用されます。

Console

ML 入力チャネルを作成するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションページで、ML 入力チャネルを作成するコラボレーションを選択します。

4. コラボレーションが開いたら、ML モデルタブを選択します。
5. カスタム ML モデル の ML 入力チャネル セクションで、ML 入力チャネルの作成 を選択します。
6. ML 入力チャネルの作成ページで、ML 入力チャネルの詳細について、次の操作を行います。
 - a. 名前 に、チャネルの一意の名前を入力します。
 - b. (オプション) 説明 に、チャネルの説明を入力します。
 - c. 関連付けられたモデルアルゴリズムで、使用するアルゴリズムを選択します。

モデルアルゴリズムを関連付けを選択して新しいアルゴリズムを追加します。

7. データセット で、トレーニングデータセットを生成する方法を選択します。
 - SQL クエリを選択して、SQL クエリの結果をトレーニングデータセットとして使用します。

SQL クエリを選択した場合は、SQL クエリフィールドにクエリを入力します。

(オプション) 最近使用したクエリをインポートするには、最近のクエリからインポートを選択します。

- 分析テンプレートを選択して、分析テンプレートの結果をトレーニングデータセットとして使用します。

Warning

合成データ生成は、特定の個人が元のデータセットに存在するか、それらの個人の学習属性が存在するかにかかわらず、個々の属性の推測から保護します。ただし、個人を特定できる情報 (PII) を含む元のデータセットのリテラル値が合成データセットに表示されるのを防ぐことはできません。

1つのデータサブジェクトのみに関連付けられている入力データセット内の値は、データセットを再識別する可能性があるため、避けることをお勧めします。たとえば、郵便番号にユーザーが1人しかいない場合、合成データセットにその郵便番号が存在すると、そのユーザーが元のデータセットに属していたことが確認されます。高精度値の切り捨てや、まれなカタログを他のカタログに置き換えるなどの手法を使用して、このリスクを軽減できます。これらの変換は、ML 入力チャネルの作成に使用されるクエリの一部にすることができます。

- a. テーブルが関連付けられていない場合は、テーブルを関連付けて、指定されたモデルに対して実行できる分析ルールでテーブルを追加します。
- b. このデータチャネルを作成するときに使用するワーカータイプを選択します。デフォルトのワーカータイプは CR.1X です。使用するワーカーの数を指定します。デフォルトのワーカー番号は 16 です。Spark プロパティを指定するには:
 - i. [Spark のプロパティ] を拡張します。
 - ii. Spark プロパティの追加を選択します。
 - iii. Spark プロパティダイアログボックスで、ドロップダウンリストからプロパティ名を選択し、値を入力します。

次の表は、各プロパティの定義を示しています。

Spark プロパティの詳細については、Apache [Spark](#) ドキュメントの「Spark Properties」を参照してください。

プロパティ名	説明	デフォルト値
spark.task.maxFailures	ジョブが失敗するまでにタスクが失敗する連続回数を制御します。1 以上の値が必要です。許可される再試行回数は、この値から 1 を引いた値に等しくなります。試行が成功すると、失敗数はリセットされます。さまざまなタスクにまたがる障害は、この制限に累積されません。	4
spark.sql.files.maxPartitionBytes	Parquet、JSON、ORC などのファイルベースのソースから読み取るときに 1 つのパーティションにパックする最大バイト数を設定します。	128MB
spark.hadoop.fs.s3.maxRetries	Amazon S3 ファイルオペレーションの再試行の最大回数を設定します。	

プロパティ名	説明	デフォルト値
spark.network.timeout	すべてのネットワークインタラクションのデフォルトのタイムアウトを設定します。設定されていない場合、次のタイムアウト設定を上書きします。 - spark.storage.blockManagerHeartbeatTimeoutMs - spark.shuffle.io.connectionTimeout - spark.rpc.askTimeout - spark.rpc.lookupTimeout	120 秒
spark.rdd.compress	spark.io.compression.codec を使用してシリアル化された RDD パーティションを圧縮するかどうかを指定します。Java および Scala の StorageLevel.MEMORY_ONLY_SER、または Python の StorageLevel.MEMORY_ONLY に適用されます。ストレージ領域を削減しますが、追加の CPU 処理時間が必要です。	FALSE
spark.shuffle.spill.compress	spark.io.compression.codec を使用してシャッフルスピルデータを圧縮するかどうかを指定します。	TRUE
spark.sql.adaptive.advisoryPartitionSizeInBytes	spark.sql.adaptive.enabled が true の場合、適応最適化中のシャッフルパーティションのターゲットサイズをバイト単位で設定します。小さなパーティションを結合するとき、または歪んだパーティションを分割するときのパーティションサイズを制御します。	(spark.sql.adaptive.shuffle.targetPostShuffleInputSize の値)

プロパティ名	説明	デフォルト値
spark.sql.adaptive .autoBroadcastJoin Threshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。適応フレームワークにのみ適用されます。spark.sql.adaptive.autoBroadcastJoinThreshold と同じデフォルト値を使用します。ブロードキャストを無効にするには、-1 に設定します。	(none)
spark.sql.adaptive .coalescePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytes に基づいて連続するシャッフルパーティションを結合してタスクサイズを最適化するかどうかを指定します。spark.sql.adaptive.enabled を true にする必要があります。	TRUE
spark.sql.adaptive .coalescePartitions.initialPartition Num	結合前のシャッフルパーティションの初期数を定義します。spark.sql.adaptive.enabled と spark.sql.adaptive.coalescePartitions.enabled の両方が true である必要があります。デフォルトは spark.sql.shuffle.partitions の値です。	(none)
spark.sql.adaptive .coalescePartitions.minPartitionSize	アダプティブ最適化中にパーティションが小さすぎるのを防ぐために、結合されたシャッフルパーティションの最小サイズを設定します。	1 MB

プロパティ名	説明	デフォルト値
spark.sql.adaptive.coalescePartitions.parallelismFirst	パーティションの結合中に spark.sql.adaptive.advisoryPartitionSizeInBytes ではなく、クラスター並列処理に基づいてパーティションサイズを計算するかどうかを指定します。並列処理を最大化するために、設定されたターゲットサイズよりも小さいパーティションサイズを生成します。これをビジー状態のクラスターで false に設定して、過剰な小さなタスクを防止することでリソース使用率を向上させることをお勧めします。	TRUE
spark.sql.adaptive.enabled	正確なランタイム統計に基づいて、アダプティブクエリの実行を有効にして、クエリの実行中にクエリプランを再最適化するかどうかを指定します。	TRUE
spark.sql.adaptive.forceOptimizeSkewedJoin	追加のシャッフルを導入した場合でも、OptimizeSkewedJoin を強制的に有効にするかどうかを指定します。	FALSE
spark.sql.adaptive.localShuffleReader.enabled	ソートマージ結合からブロードキャストハッシュ結合に変換した後など、シャッフルパーティショニングが必要ない場合にローカルシャッフルリーダーを使用するかどうかを指定します。spark.sql.adaptive.enabled を true にする必要があります。	TRUE

プロパティ名	説明	デフォルト値
<code>spark.sql.adaptive.maxShuffledHashJoinLocalMapThreshold</code>	ローカルハッシュマップを構築するための最大パーティションサイズをバイト単位で設定します。次の場合に、シャッフルされたハッシュ結合をソートマージ結合よりも優先します。 • この値は <code>spark.sql.adaptive.advisoryPartitionSizeInBytes</code> 以上 • すべてのパーティションサイズがこの制限内です <code>spark.sql.join.preferSortMergeJoin</code> 設定を上書きします。	0 バイト
<code>spark.sql.adaptive.optimizeSkewsInRebalancePartitions.enabled</code>	<code>spark.sql.adaptive.advisoryPartitionSizeInBytes</code> に基づいて小さなパーティションに分割することで、歪んだシャッフルパーティションを最適化するかどうかを指定します。<code>spark.sql.adaptive.enabled</code> を <code>true</code> にする必要があります。	TRUE
<code>spark.sql.adaptive.rebalancePartitionsSmallPartitionFactor</code>	分割中にパーティションをマージするためのサイズしきい値係数を定義します。この係数より小さいパーティションに <code>spark.sql.adaptive.advisoryPartitionSizeInBytes</code> を掛けるとマージされます。	0.2

プロパティ名	説明	デフォルト値
spark.sql.adaptive.skewJoin.enabled	スキューされたパーティションを分割してオプションでレプリケートすることで、シャッフルされた結合でデータスキューを処理するかどうかを指定します。ソートマージおよびシャッフルされたハッシュ結合に適用されます。spark.sql.adaptive.enabled を true にする必要があります。	TRUE
spark.sql.adaptive.skewJoin.skewedPartitionFactor	パーティションスキューを決定するサイズ係数を決定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - この係数をパーティションサイズの中央値で乗算 - spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes の値	5
spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes	歪んだパーティションを識別するためのサイズしきい値をバイト単位で設定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - このしきい値 - パーティションサイズの中央値に spark.sql.adaptive.skewJoin.skewedPartitionFactor を掛けた値 この値は spark.sql.adaptive.advisoryPartitionSizeInBytes よりも大きく設定することをお勧めします。	256MB

プロパティ名	説明	デフォルト値
spark.sql.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。ブロードキャストを無効にするには、-1 に設定します。	10MB
spark.sql.broadcastTimeout	ブロードキャスト結合中のブロードキャストオペレーションのタイムアウト期間を秒単位で制御します。	300 秒
spark.sql.cbo.enabled	計画統計の推定でコストベースの最適化 (CBO) を有効にするかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.star.filter	コストベースの結合列挙中にスター結合フィルターヒューリスティックを適用するかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.threshold	動的プログラミングアルゴリズムで許可される結合ノードの最大数を設定します。	12
spark.sql.cbo.joinReorder.enabled	コストベースの最適化 (CBO) で結合順序変更を有効にするかどうかを指定します。	FALSE
spark.sql.cbo.planStats.enabled	論理プランの生成中にカタログから行数と列統計を取得するかどうかを指定します。	FALSE
spark.sql.cbo.starSchemaDetection	スタースキーマ検出に基づいて結合順序変更を有効にするかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.files.maxPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最大数を設定します。初期数がこの値を超えたときにパーティションを再スケーリングします。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.files.maxRecordsPerFile	1 つのファイルに書き込むレコードの最大数を設定します。ゼロまたは負の値に設定されている場合、制限は適用されません。	0
spark.sql.files.minPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最小数を設定します。デフォルトは spark.sql.leafNodeDefaultParallelism です。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.inMemoryColumnarStorage.batchSize	列キャッシュのバッチサイズを制御します。サイズを大きくすると、メモリ使用率と圧縮が向上しますが、out-of-memoryエラーのリスクが高まります。	10000
spark.sql.inMemoryColumnarStorage.compressed	データ統計に基づいて列の圧縮コーデックを自動的に選択するかどうかを指定します。	TRUE

プロパティ名	説明	デフォルト値
spark.sql.inMemoryColumnarStorage.enableVectorizedReader	列キャッシュのベクトル化された読み取りを有効にするかどうかを指定します。	TRUE
spark.sql.legacy.allowHashOnMapType	マップタイプデータ構造でハッシュオペレーションを許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンのマップタイプの処理との互換性を維持します。	
spark.sql.legacy.allowNegativeScaleOfDecimal	10 進数型定義で負のスケール値を許可するかどうかを指定します。このレガシー設定は、負の 10 進スケールをサポートした古い Spark バージョンとの互換性を維持します。	
spark.sql.legacy.castComplexTypesToString.enabled	複雑な型を文字列にキャストするためのレガシー動作を有効にするかどうかを指定します。古い Spark バージョンのタイプ変換ルールとの互換性を維持します。	
spark.sql.legacy.charVarcharAsString	CHAR 型と VARCHAR 型を STRING 型として扱うかどうかを指定します。このレガシー設定は、古い Spark バージョンの文字列タイプの処理との互換性を提供します。	
spark.sql.legacy.createEmptyCollectionUsingStringType	文字列タイプの要素を使用して空のコレクションを作成するかどうかを指定します。このレガシー設定は、古い Spark バージョンのコレクションの初期化動作との互換性を維持します。	

プロパティ名	説明	デフォルト値
spark.sql.legacy.exponentialAsDecimal.enabled	指数リテラルを 10 進数型として解釈するかどうかを指定します。このレガシー設定は、古い Spark バージョンの数値リテラル処理との互換性を維持します。	
spark.sql.legacy.json.allowEmptyString.enabled	JSON 処理で空の文字列を許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンの JSON 解析動作との互換性を維持します。	
spark.sql.legacy.parquet.int96RebaseModelRead	Parquet ファイルの読み取り時にレガシー INT96 タイムスタンプリベースモードを使用するかどうかを指定します。このレガシー設定は、古い Spark バージョンのタイムスタンプ処理との互換性を維持します。	
spark.sql.legacy.timeParserPolicy	後方互換性のための解析動作を制御します。このレガシー設定により、文字列からタイムスタンプと日付を解析する方法が決まります。	
spark.sql.legacy.typeCoercion.datetimeToString.enabled	日時値を文字列に変換するときにレガシー型の強制動作を有効にするかどうかを指定します。古い Spark バージョンの日時変換ルールとの互換性を維持します。	
spark.sql.maxSinglePartitionBytes	最大パーティションサイズをバイト単位で設定します。プランナーは、並列処理を改善するために、大きなパーティションのシャッフルオペレーションを導入します。	128 メートル

プロパティ名	説明	デフォルト値
spark.sql.metadataCacheTTLSeconds	メタデータキャッシュtime-to-live (TTL) を制御します。パーティションファイルメタデータとセッションカタログキャッシュに適用されます。以下が必要です。 0 より大きい正の値 - spark.sql.catalogImplementationを hive に設定 - spark.sql.hive.filesourcePartitionFileCacheSize が 0 より大きい - spark.sql.hive.manageFilesourcePartitions を true に設定	-1000 ミリ秒
spark.sql.optimizer.collapseProjectAlwaysInline	重複が発生した場合でも、隣接する射影とインライン式を折りたたむかどうかを指定します。	FALSE
spark.sql.optimizer.dynamicPartitionPruning.enabled	結合キーとして使用されるパーティション列の述語を生成するかどうかを指定します。	TRUE
spark.sql.optimizer.enableCsvExpressionOptimization	from_csv オペレーションから不要な列を削除して、SQL オプティマイザの CSV 式を最適化するかどうかを指定します。	TRUE
spark.sql.optimizer.enableJsonExpressionOptimization	SQL オプティマイザで JSON 式を最適化するかどうかを次のように指定します。 - from_json オペレーションから不要な列を削除する - from_json と to_json の組み合わせの簡素化 - named_struct オペレーションの最適化	TRUE

プロパティ名	説明	デフォルト値
spark.sql.optimize r.excludedRules	無効にするオプティマイザルールを定義し、カンマ区切りのルール名で識別します。一部のルールは、正確性のために必要であるため、無効にすることはできません。オプティマイザは、どのルールが正常に無効化されたかをログに記録します。	(none)
spark.sql.optimize r.runtime.bloomFilter.applicationSideScanSizeThreshold	アプリケーション側に Bloom フィルターを挿入するために必要な最小集約スキャンサイズをバイト単位で設定します。	10GB
spark.sql.optimize r.runtime.bloomFilter.creationSideThreshold	作成側で Bloom フィルターを挿入するための最大サイズしきい値を定義します。	10MB
spark.sql.optimize r.runtime.bloomFilter.enabled	シャッフル結合の片側に選択的述語がある場合に、シャッフルデータを減らすために Bloom フィルターを挿入するかどうかを指定します。	TRUE
spark.sql.optimize r.runtime.bloomFilter.expectedNumItems	ランタイム Bloom フィルターで予想される項目のデフォルトの数を定義します。	1000000
spark.sql.optimize r.runtime.bloomFilter.maxNumBits	ランタイム Bloom フィルターで許可される最大ビット数を設定します。	67108864
spark.sql.optimize r.runtime.bloomFilter.maxNumItems	ランタイム Bloom フィルターで許可される予想項目の最大数を設定します。	4000000

プロパティ名	説明	デフォルト値
spark.sql.optimize r.runtime.bloomFilter.number.threshold	ドライバーのout-of-memoryエラーを防ぐために、クエリごとに許可される非DPP ランタイムフィルターの最大数を制限します。	10
spark.sql.optimize r.runtime.bloomFilter.numBits	ランタイム Bloom フィルターで使用するデフォルトのビット数を定義します。	8388608
spark.sql.optimize r.runtime.rowlevel OperationGroupFilter.enabled	行レベルのオペレーションでランタイムグループのフィルタリングを有効にするかどうかを指定します。データソースに以下を許可します。 - データソースフィルターを使用してデータグループ全体 (ファイルやパーティションなど) を削除する - ランタイムクエリを実行して一致するレコードを特定する - 不要なグループを破棄して、高価な書き換えを避ける 機能制限: - すべての式がデータソースフィルターに変換できるわけではありません - 一部の式では、Spark 評価 (サブクエリなど) が必要です。	TRUE
spark.sql.optimize r.runtimeFilter.semiJoinReduction.enabled	シャッフル結合の片側に選択述語がある場合に、シャッフルデータを減らすためにセミ結合を挿入するかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.parquet.aggregatePushdown	最適化のために集計を Parquet にプッシュダウンするかどうかを指定します。以下をサポートします。 • ブール型、整数型、浮動小数点型、日付型の MIN と MAX • すべてのデータ型の COUNT Parquet ファイルフッターに統計情報がない場合、例外をスローします。	FALSE
spark.sql.parquet.columnarReaderBatchSize	Parquet ベクトル化された各リーダーバッチの行数を制御します。out-of-memory エラーを防ぐために、パフォーマンスのオーバーヘッドとメモリ使用量のバランスを取る値を選択します。	4096
spark.sql.session.timeZone	文字列リテラルおよび Java オブジェクト変換のタイムスタンプを処理するセッションタイムゾーンを定義します。以下を受け入れます。 • 地域/都市形式のリージョンベースの IDs (アメリカ/Los_Angeles など) • (+/-)HH、(+/-)HH:mm、または (+/-)HH:mm:ss 形式のゾーンオフセット (-08 や +01:00 など) • +00:00 のエイリアスとしての UTC または Z	(ローカルタイムゾーンの値)
spark.sql.shuffle.partitions	結合または集約中のデータシャッフルのパーティションのデフォルト数を設定します。同じチェックポイントの場所からの構造化ストリーミングクエリの再起動の間に変更することはできません。	200

プロパティ名	説明	デフォルト値
spark.sql.shuffledHashJoinFactor	シャッフルハッシュ結合の適格性を判断するために使用される乗算係数を定義します。シャッフルハッシュ結合は、スモールサイドのデータサイズにこの係数を掛けた値がラージサイドのデータサイズより小さい場合に選択されます。	3
spark.sql.sources.parallelPartitionDiscovery.threshold	ファイルベースのソース (Parquet、JSON、ORC) を使用したドライバー側のファイルリストの最大パス数を設定します。パーティション検出中に超過すると、ファイルは別の Spark 分散ジョブを使用して一覧表示されます。	32
spark.sql.statistics.histogram.enabled	推定精度を向上させるために、列統計の計算中に等高ヒストグラムを生成するかどうかを指定します。基本的な列統計に必要な範囲を超える追加のテーブルスキャンが必要です。	FALSE

- c. データ保持日数には、データを保持する日数を入力します。
 - d. 結果形式では、ML 入力チャネルが使用するデータ形式として CSV または Parquet を選択します。
8. サービスアクセスでは、このテーブルへのアクセスに使用する既存のサービスロール名を選択するか、新しいサービスロールを作成して使用します。
 9. 暗号化では、カスタム KMS キーを使用して Encrypt シークレットを選択し、独自の KMS キーと関連情報を指定します。それ以外の場合、Clean Rooms ML が暗号化を管理します。
 10. ML 入力チャネルの作成 を選択します。

ML 入力チャネルの作成には数分かかります。ML 入力チャネルのリストは、ML モデルタブで確認できます。

Note

ML 入力チャネルを作成した後は、編集できません。

API

ML 入力チャネルを作成するには (API)

特定のパラメータで次のコードを実行します。

```
import boto3
acr_client = boto3.client('cleanroomsm1')

acr_client.create_ml_input_channel(
    name="ml_input_channel_name",
    membershipIdentifier='membership_id',

    configuredModelAlgorithmAssociations=[configured_model_algorithm_association_arn],
    retentionInDays=1,
    inputChannel={
        "dataSource": {
            "protectedQueryInputParameters": {
                "sqlParameters": {
                    "queryString": "select * from table",
                    "computeConfiguration": {
                        "worker": {
                            "type": "CR.1X",
                            "number": 16,
                            "properties": {
                                "spark": {
                                    "spark configuration key": "spark configuration
value",
                                }
                            }
                        }
                    }
                },
            },
            "resultFormat": "PARQUET"
        }
    },
    "roleArn": "arn:aws:iam::111122223333:role/role_name"
```

```
)  
channel_arn = resp['ML Input Channel ARN']
```

AWS Clean Rooms ML でのトレーニング済みモデルの作成

前提条件:

- にアクセスできる AWS アカウント 。 AWS Clean Rooms
- でのコラボレーションのセットアップ AWS Clean Rooms
- コラボレーションに関連付けられた設定済みモデルアルゴリズム
- 少なくとも 1 つの設定された ML 入力チャネル
- コラボレーションで ML モデルを作成および管理するための適切なアクセス許可

設定されたモデルアルゴリズムをコラボレーションに関連付け、ML 入力チャネルを作成して設定したら、トレーニング済みモデルを作成する準備が整います。トレーニング済みモデルは、共同でデータを分析するためにコラボレーションのメンバーによって使用されます。

次の手順を使用して、トレーニング済みモデルを作成できます。

または、増分トレーニングを使用して新しいデータで既存のモデルを改善したり、分散トレーニングを使用して複数のコンピューティングインスタンスにまたがってモデルをトレーニングしたりできます。

トピック

- [AWS Clean Rooms ML での増分トレーニングの使用](#)
- [AWS Clean Rooms ML での分散トレーニングの使用](#)

Console

トレーニング済みモデルを作成するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションページで、トレーニング済みモデルを作成するコラボレーションを選択します。

4. コラボレーションが開いたら、ML モデルタブを選択します。
5. カスタム ML モデル セクションで、トレーニング済みモデルの作成 を選択します。
6. トレーニング済みモデルの作成ページで、関連付けられたモデルアルゴリズムにアルゴリズムを指定します。
7. トレーニング済みモデルの詳細については、次のように入力します。
 - a. Name には、コラボレーションでモデルの一意の名前を入力します。
 - b. (オプション) 説明 に、トレーニングされたモデルの説明を入力します。
 - c. トレーニングデータ入力モードでは、次のいずれかを選択します。
 - ML ストレージボリュームに収まる小さなデータセットがあり、トレーニングスクリプトに従来のファイルシステムアクセスを希望する場合は、ファイルを選択します。
 - 大規模なデータセットにパイプを選択して S3 から直接データをストリーミングすると、すべてをディスクにダウンロードする必要がなくなるため、トレーニング速度が向上し、ストレージ要件が軽減されます。
 - S3 からのストリーミングの利点とファイルシステムアクセスを組み合わせる場合は、FastFile を選択します。特にデータのシーケンシャル読み取りや、起動時間の短縮のためにより少ないファイルを処理する場合は、FastFile を選択します。
8. ML 入力チャネルの詳細については、以下を実行します。
 - a. ML 入力チャネルでは、モデルアルゴリズムにデータを提供する ML 入力チャネルを指定します。

別のチャネルを追加するには、別の ML 入力チャネルを追加する を選択します。最大 19 個の ML 入力チャネルを追加できます。
 - b. チャネル名に、ML 入力チャネルの名前を入力します。
 - c. Amazon S3 データディストリビューションタイプでは、次のいずれかを選択します。
 - 完全レプリケートを選択して、各トレーニングインスタンスにデータセットの完全なコピーを提供します。これは、データセットがメモリに収まるほど小さい場合、または各インスタンスがすべてのデータにアクセスする必要がある場合に最適です。
 - S3 キーでシャーディングを選択して、S3 キーに基づいてデータセットをトレーニングインスタンス間で分割します。各インスタンスは、合計 S3 オブジェクトの約 $1/n$ を受け取ります。ここで、「 n 」はインスタンスの数です。これは、並行して処理する大規模なデータセットに最適です。

Note

ディストリビューションタイプを選択するときは、データセットのサイズとトレーニング要件を考慮してください。完全にレプリケートされた は完全なデータアクセスを提供しますが、より多くのストレージが必要です。一方、S3 キーによるシャードにより、大規模なデータセットの分散処理が可能になります。

9. 最大トレーニング期間 で、モデルをトレーニングする最大時間を選択します。
10. ハイパーパラメータには、アルゴリズム固有のパラメータとその目的の値を指定します。ハイパーパラメータは、トレーニング対象のモデルに固有であり、モデルトレーニングを微調整するために使用されます。
11. 環境変数には、アルゴリズム固有の変数とその意図した値を指定します。環境変数は Docker コンテナで設定されます。
12. 暗号化では、カスタム を使用するには AWS KMS key、カスタム KMS キーのチェックボックスで 暗号化シークレットを選択します。
13. EC2 リソース設定では、モデルトレーニングに使用されるコンピューティングリソースに関する情報を指定します。
 - a. インスタンスタイプで、実行するインスタンスのタイプを選択します。
 - b. インスタンス数には、インスタンスの数を入力します。
 - c. ボリュームサイズを GB 単位で入力するには、ML ストレージボリュームサイズを入力します。
14. トレーニング済みモデルの作成 を選択します。

API

トレーニング済みモデルを作成するには (API)

モデルをトレーニングできるメンバーは、ML 入力チャネルとモデルアルゴリズムを選択してトレーニングを開始します。

特定のパラメータで次のコードを実行します。

```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')
```

```
acr_ml_client.create_trained_model(  
    membershipIdentifier= 'membership_id',  
    configuredModelAlgorithmAssociationArn = 'arn:aws:cleanrooms-  
ml:region:account:membership/membershipIdentifier/configured-model-algorithm-  
association/identifier',  
    name='trained_model_name',  
    resourceConfig={  
        'instanceType': "ml.m5.xlarge",  
        'volumeSizeInGB': 1  
    },  
    dataChannels=[  
        {  
            "mlInputChannelArn": channel_arn_1,  
            "channelName": "channel_name"  
        },  
        {  
            "mlInputChannelArn": channel_arn_2,  
            "channelName": "channel_name"  
        }  
    ]  
)
```

Note

トレーニング済みモデルを作成した後は、編集できません。変更するには、トレーニング済みモデルを削除し、新しいモデルを作成します。

AWS Clean Rooms ML での増分トレーニングの使用

前提条件:

- にアクセスできる AWS アカウント 。 AWS Clean Rooms
- コラボレーションの既存のトレーニング済みモデル
- 増分トレーニング用の新規または更新されたデータセット
- コラボレーションで ML モデルを作成および管理するための適切なアクセス許可
- 既存のモデルのハイパーパラメータと設定に関する知識

増分トレーニングでは、既存のモデルと更新されたデータセットのアーティファクトを使用して、新しいモデルをトレーニングできます。段階的トレーニングは時間とリソースの両方を節約します。

段階的トレーニングは次の目的で使用します。

- 前のトレーニングで考慮されなかった基盤となるパターンを持つ拡張データセットを使用して、新しいモデルをトレーニングします。
- 異なるハイパーパラメータを使用するか、異なるデータセットを使用して、モデルの複数のバリエーションをトレーニングします。

Console

増分トレーニングジョブを実行するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションページで、増分トレーニングに使用するモデルアーティファクトが存在するコラボレーションを選択します。
4. コラボレーションが開いたら、ML モデルタブを選択します。
5. カスタム ML モデル の「トレーニング済みモデル」セクションで、段階的にトレーニングするトレーニング済みモデルの横にあるラジオボタンを選択します。
6. 概要ページのバージョン
 - a. 段階的にトレーニングするトレーニング済みモデルの横にあるラジオボタンを選択します。
 - b. バージョンからトレーニングを選択します。
7. 「バージョンからトレーニング済みモデルを作成する」ページの「トレーニング済みモデルバージョン」で、バージョンを選択します。

ベースモデルバージョンが自動的に選択されます。他のバージョンが存在する場合は、このバージョンを変更できます。

8. トレーニング済みモデルの詳細については、次のように入力します。
 - a. Name に、コラボレーションでモデルの一意の名前を入力します。
 - b. (オプション) 説明 に、トレーニングされたモデルの説明を入力します。

- c. トレーニングデータ入力モードでは、次のいずれかを選択します。
- ML ストレージボリュームに収まる小さなデータセットがあり、トレーニングスクリプトに従来のファイルシステムアクセスを希望する場合は、ファイルを選択します。
 - 大規模なデータセットにパイプを選択して S3 から直接データをストリーミングすると、すべてをディスクにダウンロードする必要がなくなるため、トレーニング速度が向上し、ストレージ要件が軽減されます。
 - S3 からのストリーミングの利点とファイルシステムアクセスを組み合わせる場合は、FastFile を選択します。特にデータのシーケンシャル読み取りや、起動時間の短縮のためにより少ないファイルを処理する場合は、FastFile を選択します。
- d. 増分トレーニングチャンネル名には、増分トレーニングチャンネルの名前を入力します。

 Note

バージョン ID なしで増分トレーニングチャンネル名を指定すると、システムはベースモデルを使用して増分トレーニングを行います。

9. ML 入力チャンネルの詳細については、以下を実行します。
- a. ML 入力チャンネルでは、モデルアルゴリズムにデータを提供する ML 入力チャンネルを指定します。
- 別のチャンネルを追加するには、別の ML 入力チャンネルを追加する を選択します。最大 19 個の ML 入力チャンネルを追加できます。
- b. チャンネル名に、ML 入力チャンネルの名前を入力します。
- c. Amazon S3 データディストリビューションタイプでは、次のいずれかを選択します。
- 完全にレプリケートされた を選択して、各トレーニングインスタンスにデータセットの完全なコピーを提供します。これは、データセットがメモリに収まるほど小さい場合、または各インスタンスがすべてのデータにアクセスする必要がある場合に最適です。
 - S3 キーでシャーディングを選択して、S3 キーに基づいてデータセットをトレーニングインスタンス間で分割します。各インスタンスは、合計 S3 オブジェクトの約 $1/n$ を受け取ります。ここで、「 n 」はインスタンスの数です。これは、並行して処理する大規模なデータセットに最適です。

Note

ディストリビューションタイプを選択するときは、データセットのサイズとトレーニング要件を考慮してください。完全にレプリケートされた は完全なデータアクセスを提供しますが、より多くのストレージが必要です。一方、S3 キーによるシャーディングにより、大規模なデータセットの分散処理が可能になります。

10. 最大トレーニング期間 で、モデルをトレーニングする最大時間を選択します。
11. ハイパーパラメータには、アルゴリズム固有のパラメータとその意図した値を指定します。ハイパーパラメータは、トレーニング対象のモデルに固有であり、モデルトレーニングを微調整するために使用されます。
12. 環境変数には、アルゴリズム固有の変数とその意図した値を指定します。環境変数は Docker コンテナで設定されます。
13. 暗号化でカスタム を使用するには AWS KMS key、カスタム KMS キーのチェックボックスで 暗号化シークレットを選択します。
14. EC2 リソース設定では、モデルトレーニングに使用されるコンピューティングリソースに関する情報を指定します。
 - a. インスタンスタイプで、実行するインスタスのタイプを選択します。
 - b. インスタンス数には、インスタスの数を入力します。
 - c. ボリュームサイズを GB 単位で入力するには、ML ストレージボリュームサイズを入力します。
15. 「バージョンからトレーニング済みモデルを作成する」を選択します。

API

増分トレーニングジョブ (API) を実行するには
特定のパラメータで次のコードを実行します。

```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')

acr_ml_client.create_trained_model(
    membershipIdentifier= 'membership_id',
```

```
configuredModelAlgorithmAssociationArn = 'arn:aws:cleanrooms-
ml:region:account:membership/membershipIdentifier/configured-model-algorithm-
association/identifier',
name='trained_model_name',
resourceConfig={
    'instanceType': 'ml.m5.xlarge',
    'volumeSizeInGB': 1
},
incrementalTrainingDataChannels=[
    {
        'trainedModelArn': trained_model_arn,
        'channelName': 'channel_name'
    },
]
dataChannels=[
    {
        'mlInputChannelArn': channel_arn_1,
        'channelName': 'channel_name'
    },
    {
        'mlInputChannelArn': channel_arn_2,
        'channelName': 'channel_name'
    }
]
)
```

Note

制限: 最大 20 チャネル (dataChannelsと の両方を含むincrementalTrainingDataChannels)。

Note

トレーニング済みモデルを作成した後は、編集できません。変更するには、トレーニング済みモデルを削除し、新しいモデルを作成します。

AWS Clean Rooms ML での分散トレーニングの使用

前提条件:

- にアクセスできる AWS アカウント 。 AWS Clean Rooms
- でのコラボレーションのセットアップ AWS Clean Rooms
- 分散トレーニングをサポートする設定済みモデルアルゴリズム
- 分散処理に適した大きなデータセット
- コラボレーションで ML モデルを作成および管理するための適切なアクセス許可
- 分散トレーニング用に複数のインスタンスを実行するのに十分な Amazon EC2 クォータ

分散トレーニングは、多数のコンピューティングノードの能力を活用して並行して動作し、大量のデータを処理してモデルパラメータを効率的に更新します。

分散トレーニングの詳細については、Amazon SageMakerデベロッパーガイド」の「[分散トレーニングの概念](#)」を参照してください。

Console

分散トレーニングジョブを実行するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションページで、トレーニング済みモデルを作成するコラボレーションを選択します。
4. コラボレーションが開いたら、ML モデルタブを選択します。
5. 「カスタム ML モデル」の「トレーニング済みモデルの作成」セクションで「トレーニング済みモデルの作成」を選択します。
6. トレーニング済みモデルの作成ページで、関連付けられたモデルアルゴリズムにアルゴリズムを指定します。
7. トレーニング済みモデルの詳細については、次のように入力します。
 - a. Name に、コラボレーションでモデルの一意の名前を入力します。
 - b. (オプション) 説明 に、トレーニングされたモデルの説明を入力します。
 - c. トレーニングデータ入力モードでは、次のいずれかを選択します。
 - ML ストレージボリュームに収まる小さなデータセットがあり、トレーニングスクリプトに従来のファイルシステムアクセスを希望する場合は、ファイルを選択します。

- 大規模なデータセットのパイプを選択して S3 から直接データをストリーミングするため、すべてをディスクにダウンロードする必要がなくなり、トレーニング速度が向上し、ストレージ要件が軽減されます。
- S3 からのストリーミングの利点とファイルシステムアクセスを組み合わせる場合は、FastFile を選択します。特にデータのシーケンシャル読み取りや、起動時間の短縮のためにより少ないファイルを処理する場合は、FastFile を選択します。

8. ML 入力チャネルの詳細については、以下を実行します。

- a. ML 入力チャネルでは、モデルアルゴリズムにデータを提供する ML 入力チャネルを指定します。

別のチャネルを追加するには、別の ML 入力チャネルを追加する を選択します。最大 19 個の ML 入力チャネルを追加できます。

- b. チャネル名に、ML 入力チャネルの名前を入力します。

- c. Amazon S3 データディストリビューションタイプでは、次のいずれかを選択します。

- 完全にレプリケートされた を選択して、各トレーニングインスタンスにデータセットの完全なコピーを提供します。これは、データセットがメモリに収まるほど小さい場合、または各インスタンスがすべてのデータにアクセスする必要がある場合に最適です。
- S3 キーでシャーディングを選択して、S3 キーに基づいてデータセットをトレーニングインスタンス間で分割します。各インスタンスは、合計 S3 オブジェクトの約 $1/n$ を受け取ります。ここで、「 n 」はインスタンスの数です。これは、並行して処理する大規模なデータセットに最適です。

 Note

ディストリビューションタイプを選択するときは、データセットのサイズとトレーニング要件を考慮してください。完全にレプリケートされた は完全なデータアクセスを提供しますが、より多くのストレージが必要です。S3 キーでシャードされた は、大規模なデータセットの分散処理を可能にします。

9. 最大トレーニング期間 で、モデルをトレーニングする最大時間を選択します。

10. ハイパーパラメータには、アルゴリズム固有のパラメータとその目的の値を指定します。ハイパーパラメータは、トレーニング対象のモデルに固有であり、モデルトレーニングを微調整するために使用されます。

11. 環境変数には、アルゴリズム固有の変数とその目的の値を指定します。環境変数は Docker コンテナで設定されます。
12. 暗号化でカスタム を使用するには AWS KMS key、カスタム KMS キーのチェックボックスで Encrypt シークレットを選択します。
13. EC2 リソース設定では、モデルトレーニングに使用されるコンピューティングリソースに関する情報を指定します。
 - a. インスタンスタイプで、実行するインスタンスのタイプを選択します。

分散トレーニングでサポートされているインスタンスタイプは次のとおりです。

- ml.m5.4xlarge
- ml.m5.12xlarge
- ml.m5.2xlarge
- ml.g5.12xlarge
- ml.g5.24xlarge

- b. インスタンス数には、インスタンスの数を入力します。
- c. ボリュームサイズを GB 単位で入力するには、ML ストレージボリュームサイズを入力します。

14. トレーニング済みモデルの作成 を選択します。

API

分散トレーニングジョブ (API) を実行するには

特定のパラメータで次のコードを実行します。

```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')

acr_ml_client.create_trained_model(
    membershipIdentifier= 'membership_id',
    configuredModelAlgorithmAssociationArn = 'arn:aws:cleanrooms-
ml:region:account:membership/membershipIdentifier/configured-model-algorithm-
association/identifier',
    name='trained_model_name',
    trainingInputMode: "File",
    resourceConfig={
```

```
        'instanceCount': "3",
        'instanceType': "ml.m5.xlarge",
        'volumeSizeInGB': 3
    },
    dataChannels=[
        {
            "mlInputChannelArn": channel_arn_1,
            "channelName": "channel_name",
            "S3DataDistributionType": "FullyReplicated"
        }
    ]
)
```

Note

トレーニング済みモデルを作成した後は、編集できません。変更するには、トレーニング済みモデルを削除し、新しいモデルを作成します。

AWS Clean Rooms ML からのモデルアーティファクトのエクスポート

このタスクはオプションであり、コラボレーションのメンバーにCAN_RECEIVE_MODEL_OUTPUTメンバー権限を割り当てたときに完了する必要があります。

モデルトレーニングが完了すると、モデルをトレーニングしたメンバーはモデルアーティファクトのエクスポートを開始できます。モデルをトレーニングしたメンバーは、メンバーが結果と有効な ML 設定を受信できる場合に、モデルアーティファクトを受け取るユーザーを選択します。

Console

カスタム ML モデルアルゴリズムを設定するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションページで、エクスポートするカスタムモデルを含むコラボレーションを選択します。

4. コラボレーションが開いたら、ML Models タブを選択し、カスタムトレーニング済みモデルテーブルからモデルを選択します。
5. カスタムトレーニング済みモデルの詳細ページで、モデル出力のエクスポートをクリックします。
6. モデル出力をエクスポートするには、モデル出力の詳細をエクスポートするには、名前とオプションの説明を入力します。

コラボレーションのメンバーにエクスポートされたモデル出力でモデルアーティファクトを受け取るメンバーを選択します。

7. [エクスポート] を選択します。

結果は、ML 設定で指定された Amazon S3 の場所のパス にエクスポートされます `yourSpecifiedS3Path/collaborationIdentifier/trainedModelName/callerAccountId/jobName`。エクスポートするファイルのみ、指定された最大ファイルサイズまで、設定されたモデルアルゴリズムの関連付け時に選択したファイルのみがエクスポートされます。

API

カスタム ML モデルアルゴリズム (API) を設定するには

次のコードを実行して、モデルのエクスポートを開始します。

```
import boto3
acr_ml_client= boto3.client('cleanroomsmml')

acr_ml_client.start_trained_model_export_job(
    membershipIdentifier='membership_id',
    trainedModelArn='arn:aws:cleanrooms-ml:region:account:membership/
membershipIdentifier/trained-model/identifier',
    outputConfiguration={
        'member': {
            'accountId': 'model_output_receiver_account'
        }
    },
    name='export_job_name'
)
```

結果は、ML 設定で指定された Amazon S3 の場所のパス にエクスポートされます `yourSpecifiedS3Path/collaborationIdentifier/trainedModelName/`

callerAccountId/jobName。設定済みモデルアルゴリズムの関連付け時に選択した のみをfilesToExport、maxSize指定した までエクスポートします。

AWS Clean Rooms ML でトレーニング済みモデルで推論を実行する

クエリを実行できるメンバーは、トレーニングジョブが完了すると推論ジョブを開始することもできます。推論を実行する推論データセットを選択し、推論コンテナを実行するトレーニング済みモデル出力を参照します。

推論出力を受け取るメンバーには、メンバー機能 を付与する必要がありますCAN_RECEIVE_INFERENCE_OUTPUT。

Console

モデル推論ジョブを作成するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/cleanrooms> で AWS Clean Rooms コンソールを開きます。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. コラボレーションページで、推論ジョブを作成するカスタムモデルを含むコラボレーションを選択します。
4. コラボレーションが開いたら、ML Models タブを選択し、カスタムトレーニング済みモデルテーブルからモデルを選択します。
5. カスタムトレーニング済みモデルの詳細ページで、推論ジョブの開始をクリックします。
6. 推論ジョブの開始には、推論ジョブの詳細には、名前とオプションの説明を入力します。

次の情報を入力します。

- 関連付けられたモデルアルゴリズム – 推論ジョブ中に使用される関連付けられたモデルアルゴリズム。
- ML 入力チャネルの詳細 – この推論ジョブのデータを提供する ML 入力チャネル。
- リソースの変換 – 推論ジョブの変換関数を実行するために使用されるコンピューティングインスタンス。
- 出力設定 – 推論ジョブ出力と出力の MIME タイプを受け取るユーザー。

- 暗号化 – 暗号化設定をカスタマイズを選択して、独自の KMS キーおよび関連情報を指定します。それ以外の場合、Clean Rooms ML が暗号化を管理します。
- 変換ジョブの詳細 – 推論ジョブの最大ペイロード。MB 単位。
- 環境変数 – 推論ジョブコンテナイメージにアクセスするために必要な環境変数。

7. 推論ジョブの開始 を選択します。

結果は、ML 設定で指定された Amazon S3 の場所のパス にエクスポートされます
`yourSpecifiedS3Path/collaborationIdentifier/trainedModelName/
callerAccountId/jobName`。

API

モデル推論ジョブを作成するには (API)

次のコードを実行して推論ジョブを開始します。

```
import boto3
acr_ml_client= boto3.client('cleanroomsm1')

acr_ml_client.start_trained_model_inference_job(
    name="inference_job",
    membershipIdentifier='membership_id',
    trainedModelArn='arn:aws:cleanrooms-ml:region:account:membership/  
membershipIdentifier/trained-model/identifier',

    dataSource={
        "mlInputChannelArn": 'channel_arn_3'
    },
    resourceConfig={'instanceType': 'ml.m5.xlarge'},
    outputConfiguration={
        'accept': 'text/csv',
        'members': [
            {
                "accountId": 'member_account_id'
            }
        ]
    }
)
```

結果は、ML 設定で指定された Amazon S3 の場所のパス にエクスポートされます
`yourSpecifiedS3Path/collaborationIdentifier/trainedModelName/
callerAccountId/jobName`。

トレーニングデータプロバイダーとして AWS Clean Rooms ML モデルを作成する

類似モデルとは、トレーニングデータプロバイダーのデータのモデルです。これにより、シードデータプロバイダーは、トレーニングデータプロバイダーのデータのうち、シードデータに最も近い類似セグメントを作成できます。コラボレーションで利用できる類似モデルを作成するには、トレーニングデータをインポートし、類似モデルを作成し、その類似モデルを設定して、コラボレーションに関連付ける必要があります。

類似モデルを使用するには、トレーニングデータプロバイダーとシードデータプロバイダーの2つの関係者が、データをコラボレーションに取り込む AWS Clean Rooms ために順番に で作業する必要があります。トレーニングデータプロバイダーが最初に完了しなければならないワークフローは次のとおりです。

1. トレーニングデータプロバイダーのデータは、ユーザーとアイテムのインタラクション AWS Glue のデータカタログテーブルに保存する必要があります。少なくとも、トレーニングデータにはユーザー ID 列、インタラクション ID 列、およびタイムスタンプ列が含まれている必要があります。
2. トレーニングデータプロバイダーは、トレーニングデータを に登録します AWS Clean Rooms。
3. トレーニングデータプロバイダーは、複数のシードデータプロバイダーと共有できる類似モデルを作成します。類似モデルはディープニューラルネットワークであり、トレーニングに最大 24 時間かかることがあります。このモデルは自動的に再トレーニングされないため、毎週再トレーニングすることをお勧めします。
4. トレーニングデータプロバイダーは、関連メトリクスを共有するかどうかや、出力セグメントの Amazon S3 ロケーションなど、類似モデルの設定を行います。トレーニングデータプロバイダーは、1 つの類似モデルから複数の設定済み類似モデルを作成できます。
5. トレーニングデータプロバイダーは、設定したオーディエンスモデルをシードデータプロバイダーと共有されているコラボレーションに関連付けます。

トレーニングデータプロバイダーが ML モデルの作成を完了すると、[シードデータプロバイダーは類似セグメントを作成してエクスポートできます](#)。

トピック

- [トレーニングデータのインポート](#)
- [類似モデルの作成](#)

- [類似モデルの設定](#)
- [設定された類似モデルの関連付け](#)
- [設定された類似モデルの更新](#)

トレーニングデータのインポート

Note

Amazon S3 にデータが保存されている Clean Rooms ML 類似モデルでのみ、用のトレーニングデータセットを指定できます。ただし、サポートされている任意のデータソースに保存されているデータ間で実行される SQL を使用して、類似モデルのシードデータを指定できます。

類似モデルを作成する前に、トレーニングデータを含む AWS Glue テーブルを指定する必要があります。Clean Rooms ML はこのデータのコピーを保存せず、データへのアクセスを可能にするメタデータのみを保存します。

でトレーニングデータをインポートするには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左側のナビゲーションペインで、AWS ML モデルを選択します。
3. [トレーニングデータセット] タブで [トレーニングデータセットを作成] を選択します。
4. [トレーニングデータセットを作成] ページで、[トレーニングデータセットの詳細] に [名前] と [説明] (オプション) を入力します。
5. ドロップダウンリストから、設定する [データベース] と [テーブル] を選択して、[トレーニングデータソース] を選択します。

Note

テーブルが正しいことを確認するには、次のいずれかの操作を行います。

- 「表示 AWS Glue」を選択します。
- [スキーマを表示] をオンにして、スキーマを表示します。

6. [トレーニングの詳細] で、ドロップダウンリストから [ユーザー識別子列]、[項目識別子列]、[タイムスタンプ列] を選択します。トレーニングデータにはこれら 3 つの列が含まれている必要があります。また、トレーニングデータに含める他の列を選択できます。

[タイムスタンプ列] のデータは、Unix エポック時間 (秒単位) である必要があります。

7. (オプション) [トレーニングするその他の列] がある場合は、ドロップダウンリストから [列名] と [タイプ] を選択します。
8. [サービスアクセス] で、データにアクセスできるサービスロールを指定し、データが暗号化されている場合は KMS キーを提供する必要があります。[新しいサービスロールを作成して使用] を選択すると、Clean Rooms ML は自動的にサービスロールを作成し、必要なアクセス許可ポリシーを追加します。[既存のサービスロールを使用] を選択し、使用する特定のサービスロールがある場合は、[サービスロール名] フィールドにそのサービスロールを入力します。

データが暗号化されている場合は、AWS KMS key フィールドに KMS キーを入力するか、[AWS KMS keyを作成] をクリックして新しい KMS キーを生成します。

9. トレーニングデータセットでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
10. [トレーニングデータセットを作成] を選択します。

対応する API アクションについては、「[CreateTrainingDataset](#)」を参照してください。

類似モデルの作成

トレーニングデータセットを作成したら、類似モデルを作成する準備が整います。1 つのトレーニングデータセットから多数の類似モデルを作成できます。

でデフォルトのデータベースを作成する AWS Glue Data Catalog か、指定されたロールに `aws:glue:createDatabase` 許可を含める必要があります。

で類似モデルを作成するには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左側のナビゲーションペインで、AWS ML モデルを選択します。
3. [そっくりなモデル] タブで、[類似モデルの作成] を選択します。
4. [類似モデルの作成] ページで、[類似モデルの詳細] に [名前] と [説明] (オプション) を入力します。

- a. モデル化する [トレーニングデータセット] をドロップダウンリストから選択します。

 Note

これが正しいトレーニングデータセットであることを検証するには、[トレーニングデータセットの詳細を表示] をオンにして詳細を表示します。
新しいトレーニングデータセットを作成する場合は、[トレーニングデータセットを作成] を選択します。

- b. (オプション) [トレーニングウィンドウ] に入力します。
5. 類似モデルのカスタム暗号化設定を有効にする場合は、[暗号化の設定をカスタマイズ] を選択し、KMS キーを入力します。
6. 類似モデルでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
7. [類似モデルの作成] を選択します。

 Note

モデルトレーニングには数時間から 2 日かかる場合があります。

対応する API アクションについては、「[CreateAudienceModel](#)」を参照してください。

類似モデルの設定

類似モデルを作成したら、コラボレーションで使用するように設定する準備が整います。1 つの類似モデルから、複数の設定済みの類似モデルを作成できます。

で類似モデルを設定するには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左側のナビゲーションペインで、AWS ML モデルを選択します。
3. [設定済みの類似モデル] タブで、[類似モデルの設定] を選択します。
4. [類似モデルの設定] ページで、[設定済みの類似モデルの詳細] に [名前] と [説明] (オプション) を入力します。

- a. 設定する [類似モデル] をドロップダウンリストから選択します。

 Note

これが正しい類似モデルであることを検証するには、[類似モデルの詳細を表示] をオンにして詳細を表示します。

新しい類似モデルを作成するには、[類似モデルの作成] を選択します。

- b. 希望する [マッチングシードサイズの最小値] を選択します。これは、シードデータプロバイダーのデータに含まれるユーザーのうち、トレーニングデータ内のユーザーと重複するユーザーの最小数です。この値は 0 より大きくなければなりません。
5. [他のメンバーと共有するメトリクス] では、コラボレーションのシードデータプロバイダーに関連性スコアを含むモデルメトリクスを受信させるかどうかを選択します。
6. Lookalike セグメントの送信先の場所には、類似セグメントがエクスポートされる Amazon S3 バケットを入力します。このバケットは、他のリソースと同じリージョンに存在する必要があります。
7. [サービスアクセス] では、このテーブルへのアクセスに使用する [既存のサービスロール名] を選択します。
8. 高度なビンサイズ設定では、対象者サイズタイプを絶対数またはパーセンテージとして指定します。
9. 設定済みテーブルのリソースでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
10. [類似モデルの設定] を選択します。

対応する API アクションについては、「[CreateConfiguredAudienceModel](#)」を参照してください。

設定された類似モデルの関連付け

類似モデルを設定したら、それをコラボレーションに関連付けることができます。

で設定された類似モデルを関連付けるには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。

3. [アクティブメンバーシップあり] タブでコラボレーションを選択します。
4. ML モデルタブの Ready-to-use 類似モデルで、類似モデルの関連付けを選択します。
5. [設定済みの類似モデルを関連付ける] ページの [設定済みの類似モデルの関連付け詳細] で、次の操作を行います。
 - a. 関連する設定済みオーディエンスモデルの [名前] を入力します。
 - b. テーブルの [説明] を入力します。

この説明は、似たような名前を持つ他の関連する設定済みオーディエンスモデルと区別するのに役立ちます。

6. [設定済みの類似モデル] では、ドロップダウンリストから設定済みの類似モデルを選択します。
7. 関連付ける を選択してください。

対応する API アクションについては、「[CreateConfiguredAudienceModelAssociation](#)」を参照してください。

設定された類似モデルの更新

類似モデルを関連付けたら、それを更新して、名前、共有するメトリクス、Amazon S3 の場所の出力などの情報を変更できます。

で関連付けられた設定済み類似モデルを更新するには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左側のナビゲーションペインで、AWS ML モデルを選択します。
3. [設定済みの類似モデル] タブの [すぐに使える類似モデル] で、設定済みの類似モデルを選択し、[編集] を選択します。
4. [編集] ページの [設定済みの類似モデルの関連付け詳細] で、次の操作を行います。
 - a. [名前] と [説明] (オプション) を更新します。
 - b. 設定する [類似モデル] をドロップダウンリストから選択します。
 - c. 希望する [マッチングシードサイズの最小値] を選択します。これは、シードデータプロバイダーのデータに含まれるユーザーのうち、トレーニングデータ内のユーザーと重複するユーザーの最小数です。この値は 0 より大きくなければなりません。

5. [他のメンバーと共有するメトリクス] では、コラボレーションのシードデータプロバイダーに関連性スコアを含むモデルメトリクスを受信させるかどうかを選択します。
6. [類似セグメントの送信先ロケーション] には、類似セグメントがエクスポートされる Amazon S3 バケットを入力します。このバケットは、他のリソースと同じリージョンに存在する必要があります。
7. [サービスアクセス] では、このテーブルへのアクセスに使用する [既存のサービスロール名] を選択します。
8. [高度なビンサイズの設定] で、オーディエンスのビンサイズを設定する方法を選択します。
9. [Save changes] (変更の保存) をクリックします。

対応する API アクションについては、「[UpdateConfiguredAudienceModel](#)」を参照してください。

シードデータプロバイダーとしての AWS Clean Rooms ML モデルの作成

トレーニングデータプロバイダーが ML モデルの作成を完了すると、シードデータプロバイダーは類似セグメントを作成してエクスポートできます。類似セグメントは、シードデータに最も近いトレーニングデータのサブセットです。

これは、シードデータプロバイダーが完了する必要があるワークフローです。

1. シードデータプロバイダーのデータは Amazon S3 バケットに保存することも、クエリの結果から取得することもできます。
2. シードデータプロバイダーは、トレーニングデータプロバイダーと共有するコラボレーションを開きます。
3. シードデータプロバイダーは、コラボレーションページの [Clean Rooms ML] タブから類似セグメントを作成します。
4. シードデータプロバイダーは、関連性メトリクスが共有されていれば、それを評価し、AWS Clean Roomsの外部で使用するために類似セグメントをエクスポートできます。

トピック

- [類似セグメントの作成](#)
- [類似セグメントのエクスポート](#)

類似セグメントの作成

Note

Amazon S3 にデータが保存されている Clean Rooms ML 類似モデルでのみ、用のトレーニングデータセットを指定できます。ただし、サポートされている任意のデータソースに保存されているデータ間で実行される SQL を使用して、類似モデルのシードデータを指定できます。

類似セグメントは、シードデータに最も近いトレーニングデータのサブセットです。

で類似セグメントを作成するには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます
AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [アクティブメンバーシップあり] タブでコラボレーションを選択します。
4. ML Models タブで、類似セグメントの作成を選択します。
5. 類似セグメントの作成ページで、関連付けられた設定済み類似モデルで、この類似セグメントに
使用する関連する設定済み類似モデルを選択します。
6. [類似セグメントの詳細] で、[名前] と [説明] (オプション) を入力します。
7. [シードプロファイル] で、オプションを選んで [シードメソッド] を選択し、推奨アクションを実
行します。

オプション	推奨されるアクション
Amazon S3 パス	1. Amazon S3 の場所を選択します。 2. (オプション) [シードプロファイルを出力 に含める] を選択します。
SQL クエリ	SQL クエリを記述し、その結果をシード データとして使用します。
分析テンプレート	ドロップダウンリストから分析テンプレート を選択し、分析テンプレートによって作成さ れた結果を使用します。

8. このデータソースを作成するときに使用するワーカータイプを選択します。デフォルトのワ
ーカータイプは CR.1X です。使用するワーカーの数を指定します。デフォルトはワーカー番号 16
です。Spark プロパティを指定するには:
 - a. [Spark のプロパティ] を拡張します。
 - b. Spark プロパティの追加を選択します。
 - c. Spark プロパティダイアログボックスで、ドロップダウンリストからプロパティ名を選択
し、値を入力します。

次の表は、各プロパティの定義を示しています。

Spark プロパティの詳細については、Apache [Spark](#) ドキュメントの「Spark Properties」を参照してください。

プロパティ名	説明	デフォルト値
spark.task.maxFailures	ジョブが失敗するまでにタスクが失敗する連続回数を制御します。1 以上の値が必要です。許可される再試行回数は、この値から 1 を引いた値に等しくなります。試行が成功すると、失敗回数がリセットされます。さまざまなタスクにまたがる障害は、この制限に累積されません。	4
spark.sql.files.maxPartitionBytes	Parquet、JSON、ORC などのファイルベースのソースから読み取るときに 1 つのパーティションにパックする最大バイト数を設定します。	128MB
spark.hadoop.fs.s3.maxRetries	Amazon S3 ファイルオペレーションの再試行の最大回数を設定します。	
spark.network.timeout	すべてのネットワークインタラクションのデフォルトのタイムアウトを設定します。設定されていない場合、次のタイムアウト設定を上書きします。 - spark.storage.blockManagerHeartbeatTimeoutMs - spark.shuffle.io.connectionTimeout - spark.rpc.askTimeout - spark.rpc.lookupTimeout	120 秒

プロパティ名	説明	デフォルト値
spark.rdd.compress	spark.io.compression.codec を使用してシリアル化された RDD パーティションを圧縮するかどうかを指定します。Java および Scala の StorageLevel.MEMORY_ONLY_SER、または Python の StorageLevel.MEMORY_ONLY に適用されます。ストレージ領域を削減しますが、追加の CPU 処理時間が必要です。	FALSE
spark.shuffle.spill.compress	spark.io.compression.codec を使用してシャッフルスピルデータを圧縮するかどうかを指定します。	TRUE
spark.sql.adaptive.advisoryPartitionSizeInBytes	spark.sql.adaptive.enabled が true の場合、適応最適化中のシャッフルパーティションのターゲットサイズをバイト単位で設定します。小さなパーティションを結合するとき、または歪んだパーティションを分割するときのパーティションサイズを制御します。	(spark.sql.adaptive.shuffle.targetPostShuffleInputSize の値)
spark.sql.adaptive.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。適応フレームワークにのみ適用されます。spark.sql.autoBroadcastJoinThreshold と同じデフォルト値を使用します。ブロードキャストを無効にするには、-1 に設定します。	(none)
spark.sql.adaptive.coalescePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytes に基づいて連続したシャッフルパーティションを結合してタスクサイズを最適化するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE

プロパティ名	説明	デフォルト値
spark.sql.adaptive.coalescePartitions.initialPartitionNum	結合前のシャッフルパーティションの初期数を定義します。spark.sql.adaptive.enabled と spark.sql.adaptive.coalescePartitions.enabled の両方が true である必要があります。デフォルトは spark.sql.shuffle.partitions の値です。	(none)
spark.sql.adaptive.coalescePartitions.minPartitionSize	アダプティブ最適化中にパーティションが小さすぎないように、結合されたシャッフルパーティションの最小サイズを設定します。	1 MB
spark.sql.adaptive.coalescePartitions.parallelismFirst	パーティションの結合中に spark.sql.adaptive.advisoryPartitionSizeInBytes ではなく、クラスターの並列処理に基づいてパーティションサイズを計算するかどうかを指定します。並列処理を最大化するために、設定されたターゲットサイズよりも小さいパーティションサイズを生成します。これをビジー状態のクラスターで false に設定して、過剰な小さなタスクを防止することでリソース使用率を向上させることをお勧めします。	TRUE
spark.sql.adaptive.enabled	正確なランタイム統計に基づいて、アダプティブクエリの実行を有効にして、クエリの実行中にクエリプランを再最適化するかどうかを指定します。	TRUE
spark.sql.adaptive.forceOptimizeSkewedJoin	追加のシャッフルを導入した場合でも、OptimizeSkewedJoin を強制的に有効にするかどうかを指定します。	FALSE

プロパティ名	説明	デフォルト値
spark.sql.adaptive.localShuffleReader.enabled	ソートマージ結合からブロードキャストハッシュ結合に変換した後など、シャッフルパーティショニングが必要ない場合にローカルシャッフルリーダーを使用するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.maxShuffledHashJoinLocalMapThreshold	ローカルハッシュマップを構築するための最大パーティションサイズをバイト単位で設定します。次の場合に、シャッフルされたハッシュ結合をソートマージ結合よりも優先します。 - この値は spark.sql.adaptive.advisoryPartitionSizeInBytes 以上です - すべてのパーティションサイズがこの制限内です spark.sql.join.preferSortMergeJoin 設定を上書きします。	0 バイト
spark.sql.adaptive.optimizeSkewsInRebalancePartitions.enabled	spark.sql.adaptive.advisoryPartitionSizeInBytes に基づいて小さなパーティションに分割することで、歪んだシャッフルパーティションを最適化するかどうかを指定します。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.rebalancePartitionsSmallPartitionFactor	分割中にパーティションをマージするためのサイズしきい値係数を定義します。この係数より小さいパーティションに spark.sql.adaptive.advisoryPartitionSizeInBytes を掛けるとマージされます。	0.2

プロパティ名	説明	デフォルト値
spark.sql.adaptive.skewJoin.enabled	スキューされたパーティションを分割してオプションでレプリケートすることで、シャッフルされた結合でデータスキューを処理するかどうかを指定します。ソートマージおよびシャッフルされたハッシュ結合に適用されます。spark.sql.adaptive.enabled が true である必要があります。	TRUE
spark.sql.adaptive.skewJoin.skewedPartitionFactor	パーティションスキューを決定するサイズ係数を決定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - この係数にパーティションサイズの中央値を乗算 - spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes の値	5
spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes	歪んだパーティションを識別するためのサイズしきい値をバイト単位で設定します。パーティションのサイズが両方を超えると、パーティションが歪みます。 - このしきい値 - パーティションサイズの中央値に spark.sql.adaptive.skewJoin.skewedPartitionFactor を掛けた値 この値は spark.sql.adaptive.advisoryPartitionSizeInBytes よりも大きく設定することをお勧めします。	256MB

プロパティ名	説明	デフォルト値
spark.sql.autoBroadcastJoinThreshold	結合中にワーカーノードにブロードキャストするための最大テーブルサイズをバイト単位で設定します。ブロードキャストを無効にするには、-1 に設定します。	10MB
spark.sql.broadcastTimeout	ブロードキャスト結合中のブロードキャストオペレーションのタイムアウト期間を秒単位で制御します。	300 秒
spark.sql.cbo.enabled	計画統計の推定でコストベースの最適化 (CBO) を有効にするかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.star.filter	コストベースの結合列挙中にスター結合フィルターヒューリスティックを適用するかどうかを指定します。	FALSE
spark.sql.cbo.joinReorder.dp.threshold	動的プログラミングアルゴリズムで許可される結合ノードの最大数を設定します。	12
spark.sql.cbo.joinReorder.enabled	コストベースの最適化 (CBO) で結合順序変更を有効にするかどうかを指定します。	FALSE
spark.sql.cbo.planStats.enabled	論理計画の生成中にカタログから行数と列統計を取得するかどうかを指定します。	FALSE
spark.sql.cbo.starSchemaDetection	スタースキーマ検出に基づいて結合順序変更を有効にするかどうかを指定します。	FALSE
spark.sql.files.maxPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最大数を設定します。初期数がこの値を超えたときにパーティションを再スケーリングします。これは推奨されるターゲットであり、保証された制限ではありません。	(none)

プロパティ名	説明	デフォルト値
spark.sql.files.maxRecordsPerFile	1つのファイルに書き込むレコードの最大数を設定します。ゼロまたは負の値に設定されている場合、制限は適用されません。	0
spark.sql.files.minPartitionNum	ファイルベースのソース (Parquet、JSON、ORC) の分割ファイルパーティションのターゲット最小数を設定します。デフォルトは spark.sql.leafNodeDefaultParallelism です。これは推奨されるターゲットであり、保証された制限ではありません。	(none)
spark.sql.inMemoryColumnarStorage.batchSize	列キャッシュのバッチサイズを制御します。サイズを大きくすると、メモリ使用率と圧縮が向上しますが、out-of-memoryエラーのリスクが高まります。	10000
spark.sql.inMemoryColumnarStorage.compressed	データ統計に基づいて列の圧縮コーデックを自動的に選択するかどうかを指定します。	TRUE
spark.sql.inMemoryColumnarStorage.enableVectorizedReader	列キャッシュのベクトル化された読み取りを有効にするかどうかを指定します。	TRUE
spark.sql.legacy.allowHashOnMapType	マップタイプデータ構造でハッシュオペレーションを許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンのマップタイプの処理との互換性を維持します。	
spark.sql.legacy.allowNegativeScaleOfDecimal	10 進数型定義で負のスケール値を許可するかどうかを指定します。このレガシー設定は、負の 10 進スケールをサポートした古い Spark バージョンとの互換性を維持します。	

プロパティ名	説明	デフォルト値
spark.sql.legacy.castComplexTypesToString.enabled	複雑な型を文字列にキャストするためのレガシー動作を有効にするかどうかを指定します。古い Spark バージョンのタイプ変換ルールとの互換性を維持します。	
spark.sql.legacy.charVarcharAsString	CHAR 型と VARCHAR 型を STRING 型として扱うかどうかを指定します。このレガシー設定は、古い Spark バージョンの文字列タイプの処理との互換性を提供します。	
spark.sql.legacy.createEmptyCollectionUsingStringType	文字列タイプの要素を使用して空のコレクションを作成するかどうかを指定します。このレガシー設定は、古い Spark バージョンのコレクションの初期化動作との互換性を維持します。	
spark.sql.legacy.exponentLiteralAsDecimal.enabled	指数リテラルを 10 進数型として解釈するかどうかを指定します。このレガシー設定は、古い Spark バージョンの数値リテラル処理との互換性を維持します。	
spark.sql.legacy.json.allowEmptyString.enabled	JSON 処理で空の文字列を許可するかどうかを指定します。このレガシー設定は、古い Spark バージョンの JSON 解析動作との互換性を維持します。	
spark.sql.legacy.parquet.int96RebaseModelRead	Parquet ファイルの読み取り時にレガシー INT96 タイムスタンプリベースモードを使用するかどうかを指定します。このレガシー設定は、古い Spark バージョンのタイムスタンプ処理との互換性を維持します。	
spark.sql.legacy.timeParserPolicy	後方互換性のための解析動作を制御します。このレガシー設定により、文字列からタイムスタンプと日付を解析する方法が決まります。	

プロパティ名	説明	デフォルト値
spark.sql.legacy.typeCoercion.datetimeToString.enabled	日時値を文字列に変換するときにレガシー型の強制動作を有効にするかどうかを指定します。古い Spark バージョンの日時変換ルールとの互換性を維持します。	
spark.sql.maxSinglePartitionBytes	最大パーティションサイズをバイト単位で設定します。プランナーは、並列処理を改善するために、大きなパーティションのシャッフルオペレーションを導入します。	128 メートル
spark.sql.metadataCacheTTLSeconds	メタデータキャッシュ time-to-live (TTL) を制御します。パーティションファイルメタデータとセッションカタログキャッシュに適用されます。以下が必要です。 0 より大きい正の値 - spark.sql.catalogImplementation を hive に設定 - spark.sql.hive.filesourcePartitionFileCacheSize が 0 より大きい - spark.sql.hive.manageFilesourcePartitions を true に設定	-1000 ミリ秒
spark.sql.optimizer.collapseProjectAlwaysInline	重複が発生した場合でも、隣接する射影とインライン式を折りたたむかどうかを指定します。	FALSE
spark.sql.optimizer.dynamicPartitionPruning.enabled	結合キーとして使用されるパーティション列の述語を生成するかどうかを指定します。	TRUE
spark.sql.optimizer.enableCsvExpressionOptimization	from_csv オペレーションから不要な列を削除して、SQL オプティマイザの CSV 式を最適化するかどうかを指定します。	TRUE

プロパティ名	説明	デフォルト値
spark.sql.optimize r.enableJsonExpressionOptimization	SQL オプティマイザで JSON 式を最適化するかどうかを次のように指定します。 - from_json オペレーションから不要な列を削除する - from_json と to_json の組み合わせの簡素化 - named_struct オペレーションの最適化	TRUE
spark.sql.optimize r.excludedRules	無効にするオプティマイザルールを定義し、カンマ区切りのルール名で識別します。一部のルールは、正確性のために必要であるため、無効にすることはできません。オプティマイザは、どのルールが正常に無効化されたかをログに記録します。	(none)
spark.sql.optimize r.runtime.bloomFilter.applicationSideScanSizeThreshold	アプリケーション側に Bloom フィルターを挿入するために必要な最小集約スキャンサイズをバイト単位で設定します。	10GB
spark.sql.optimize r.runtime.bloomFilter.creationSideThreshold	作成側で Bloom フィルターを挿入するための最大サイズのしきい値を定義します。	10MB
spark.sql.optimize r.runtime.bloomFilter.enabled	シャッフル結合の片側に選択的述語がある場合に、シャッフルデータを減らすために Bloom フィルターを挿入するかどうかを指定します。	TRUE
spark.sql.optimize r.runtime.bloomFilter.expectedNumItems	ランタイム Bloom フィルターで予想される項目のデフォルトの数を定義します。	1000000

プロパティ名	説明	デフォルト値
spark.sql.optimize r.runtime.bloomFilter.maxNumBits	ランタイム Bloom フィルターで許可される最大ビット数を設定します。	67108864
spark.sql.optimize r.runtime.bloomFilter.maxNumItems	ランタイム Bloom フィルターで許可される予想項目の最大数を設定します。	4000000
spark.sql.optimize r.runtime.bloomFilter.number.threshold	ドライバーのout-of-memoryエラーを防ぐために、クエリごとに許可される非 DPP ランタイムフィルターの最大数を制限します。	10
spark.sql.optimize r.runtime.bloomFilter.numBits	ランタイム Bloom フィルターで使用するデフォルトのビット数を定義します。	8388608
spark.sql.optimize r.runtime.rowlevel OperationGroupFilter.enabled	行レベルのオペレーションでランタイムグループのフィルタリングを有効にするかどうかを指定します。データソースに以下を許可します。 - データソースフィルターを使用してデータグループ全体 (ファイルやパーティションなど) を削除する - ランタイムクエリを実行して一致するレコードを特定する - 不要なグループを破棄して、高価な書き換えを避ける 機能制限: - すべての式がデータソースフィルターに変換できるわけではありません - 一部の式では Spark 評価 (サブクエリなど) が必要です	TRUE

プロパティ名	説明	デフォルト値
spark.sql.optimize r.runtimeFilter.se miJoinReduction.en abled	シャッフル結合の片側に選択的述語がある場合に、シャッフルデータを減らすためにセミ結合を挿入するかどうかを指定します。	FALSE
spark.sql.parquet. aggregatePushdown	最適化のために集計を Parquet にプッシュダウンするかどうかを指定します。以下をサポートします。 • ブール型、整数型、浮動小数点型、日付型の MIN および MAX • すべてのデータ型の COUNT Parquet ファイルフッターに統計がない場合、例外をスローします。	FALSE
spark.sql.parquet. columnarReaderBatchSize	Parquet ベクトル化された各リーダーバッチの行数を制御します。out-of-memoryエラーを防ぐために、パフォーマンスのオーバーヘッドとメモリ使用量のバランスを取る値を選択します。	4096
spark.sql.session. timeZone	文字列リテラルおよび Java オブジェクト変換のタイムスタンプを処理するセッションタイムゾーンを定義します。以下を受け入れます。 • 地域/都市形式のリージョンベースの IDs (アメリカ/Los_Angeles など) • (+/-)HH、(+/-)HH:mm、または (+/-)HH:mm:ss 形式のゾーンオフセット (-08 や +01:00 など) • +00:00 のエイリアスとしての UTC または Z	(ローカルタイムゾーンの値)

プロパティ名	説明	デフォルト値
spark.sql.shuffle.partitions	結合または集約中のデータシャッフルのパーティションのデフォルト数を設定します。同じチェックポイントの場所からの構造化ストリーミングクエリの再起動の間に変更することはできません。	200
spark.sql.shuffledHashJoinFactor	シャッフルハッシュ結合の適格性を判断するために使用される乗算係数を定義します。シャッフルハッシュ結合は、スモールサイドのデータサイズにこの係数を掛けた値がラージサイドのデータサイズより小さい場合に選択されます。	3
spark.sql.sources.parallelPartitionDiscovery.threshold	ファイルベースのソース (Parquet、JSON、ORC) を使用したドライバー側のファイルリストの最大パス数を設定します。パーティション検出中に超過すると、ファイルは別の Spark 分散ジョブを使用して一覧表示されます。	32
spark.sql.statistics.histogram.enabled	推定精度を向上させるために、列統計の計算中に等高ヒストグラムを生成するかどうかを指定します。基本的な列統計に必要な範囲を超える追加のテーブルスキャンが必要です。	FALSE

9. [サービスアクセス] では、このテーブルへのアクセスに使用する [既存のサービスロール名] を選択します。
10. トレーニングデータセットでタグを有効にする場合は、[新しいタグを追加] を選択し、キーと値のペアを入力します。
11. [類似セグメントの作成] を選択します。

対応する API アクションについては、「[StartAudienceGenerationJob](#)」を参照してください。

類似セグメントのエクスポート

類似セグメントを作成したら、そのデータを Amazon S3 バケットにエクスポートできます。

で類似セグメントをエクスポートするには AWS Clean Rooms

1. にサインイン AWS マネジメントコンソール し、 で [AWS Clean Rooms コンソール](#) を開きます AWS アカウント (まだ開いていない場合)。
2. 左のナビゲーションペインで、[コラボレーション] を選択します。
3. [アクティブメンバーシップあり] タブでコラボレーションを選択します。
4. ML Models タブで、類似セグメントを選択し、エクスポートを選択します。
5. [類似モデルのエクスポート] で、[類似モデルの詳細のエクスポート] に [名前] と [説明] (オプション) を入力します。
6. [セグメントサイズ] では、エクスポートするセグメントのサイズを選択します。
7. [エクスポート] を選択します。

対応する API アクションについては、「[StartAudienceExportJob](#)」を参照してください。

トラブルシューティング AWS Clean Rooms

このセクションでは、を使用する際に発生する可能性がある一般的な問題 AWS Clean Rooms とその修正方法について説明します。

問題

- [クエリが参照する 1 つ以上のテーブルに、関連付けられたサービスロールでアクセスできない。テーブル/ロールの所有者が、サービスロールにテーブルへのアクセス許可を付与する必要がある。](#)
- [基になるデータセットの 1 つに、サポートされていないファイル形式が使用されている。](#)
- [Cryptographic Computing for Clean Rooms の使用時に、期待どおりのクエリ結果が得られない。](#)
- [AWS Clean Rooms Spark SQL: パーティションデータがありません](#)

クエリが参照する 1 つ以上のテーブルに、関連付けられたサービスロールでアクセスできない。テーブル/ロールの所有者が、サービスロールにテーブルへのアクセス許可を付与する必要がある。

- サービスロールのアクセス許可が必要に応じて設定されていることを確認します。詳細については、「[セットアップ AWS Clean Rooms](#)」を参照してください。

基になるデータセットの 1 つに、サポートされていないファイル形式が使用されている。

- データセットがサポートされているファイル形式のいずれかであることを確認します。
 - Parquet
 - RCFile
 - TextFile
 - SequenceFile
 - RegexSerde
 - OpenCSV
 - AVRO

- JSON

詳細については、「[のデータ形式 AWS Clean Rooms](#)」を参照してください。

Cryptographic Computing for Clean Rooms の使用時に、期待どおりのクエリ結果が得られない。

Cryptographic Computing for Clean Rooms (C3R) を使用している場合は、暗号化された列がクエリで正しく使用されていることを確認します。

- sealed列は SELECT 句でのみ使用されます。
- fingerprint列は JOIN 句 (および特定の条件下では GROUP BY 句) でのみ使用されます。
- コラボレーション設定で、同じ名前のfingerprint列でしかJOINingを実行できないように指定されている場合があります。

詳細については、「[the section called “暗号コンピューティング”](#)」および「[the section called “列タイプ”](#)」を参照してください。

AWS Clean Rooms Spark SQL: パーティションデータがありません

内のすべてのパーティション AWS Glue Data Catalog にも S3 のデータが必要です。エンジンは Spark 設定を使用します。spark.sql.files.ignoreMissingFiles=False

詳細については、<https://spark.apache.org/docs/latest/sql-data-sources-generic-options.html#ignore-missing-files> を参照してください。

このエラーが発生すると、次のエラーメッセージが表示されます。 "Missing partition data: One of the configured tables is partitioned and one or more of the partitions does not have data".

Amazon S3 に存在するデータと、テーブル AWS Glue Data Catalog の にリストされているパーティションを比較します。S3 で対応するデータがないパーティションを削除します。

のセキュリティ AWS Clean Rooms

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を活用できます。

セキュリティは、AWS お客様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 クラウドで 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 AWS 責任があります。AWS また、では、安全に使用できるサービスも提供しています。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。が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 AWS Clean Rooms、「[コンプライアンスプログラムによる AWS 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。
- クラウド内のセキュリティ – 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、お客様は、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、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 AWS Clean Rooms。セキュリティとコンプライアンスの目的を達成する AWS Clean Rooms ように を設定する方法を示します。また、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 AWS Clean Rooms 説明します。

内容

- [でのデータ保護 AWS Clean Rooms](#)
- [のサービスにリンクされたロールの使用 AWS Clean Rooms](#)
- [のデータ保持 AWS Clean Rooms](#)
- [でのデータコラボレーションのベストプラクティス AWS Clean Rooms](#)
- [の Identity and Access Management AWS Clean Rooms](#)
- [のコンプライアンス検証 AWS Clean Rooms](#)
- [の耐障害性 AWS Clean Rooms](#)
- [のインフラストラクチャセキュリティ AWS Clean Rooms](#)
- [インターフェイスエンドポイント \(AWS PrivateLink\) AWS Clean Rooms を使用して AWS Clean Rooms または ML にアクセスする](#)

でのデータ保護 AWS Clean Rooms

責任 AWS [共有モデル](#)、でのデータ保護に適用されます AWS Clean Rooms。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、使用する「AWS のサービス」のセキュリティ設定と管理タスクもユーザーの責任となります。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に関するよくある質問](#)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](#)」のブログ記事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 IAM アイデンティティセンターまたは 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権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします：

- 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 は必須ですが、TLS 1.3 を推奨します。
- 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。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「AWS CloudTrail ユーザーガイド」の[CloudTrail 証跡の使用](#)を参照してください。
- 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。
- Amazon Macie などの高度な管理されたセキュリティサービスを使用します。これらは、Amazon S3 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。
- 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 140-3 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 \(FIPS\) 140-3](#)」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、コンソール AWS Clean Rooms、API、または SDK を使用して AWS CLI または他の AWS のサービス 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。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

保管中の暗号化

AWS Clean Rooms は、追加の設定を必要とせずに、保管中のすべてのサービスメタデータを常に暗号化します。この暗号化は、 の使用時に自動的に行われます AWS Clean Rooms。

Clean Rooms ML は、保管中のサービスに保存されているすべてのデータを で暗号化します AWS KMS。独自の KMS キーを提供することを選択した場合、類似モデルと類似セグメント生成ジョブの内容は KMS キーで保存時に暗号化されます。

AWS Clean Rooms カスタム ML モデルを使用する場合、サービスは保管時に保存されているすべてのデータを で暗号化します AWS KMS。 は、ユーザーが作成、所有、管理する対称カスタマーマネージドキーを使用して、保管中のデータを暗号化すること AWS Clean Rooms をサポートします。カスタマーマネージドキーが指定されていない場合は、デフォルトで AWS 所有のキー が使用されます。

AWS Clean Rooms は、権限とキーポリシーを使用してカスタマーマネージドキーにアクセスします。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 Clean Rooms 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、AWS Clean Rooms がアクセスできない暗号化された ML 入力チャネルからトレーニング済みモデルを作成しようとする、 オペレーションはValidationExceptionエラーを返します。

Note

Amazon S3 の暗号化オプションを使用して、保管中のデータを保護できます。
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 \(SSE-S3\) の指定](#)」を参照してください。

内で ID マッピングテーブルを使用する場合 AWS Clean Rooms、サービスは保管時に保存されているすべてのデータを で暗号化します AWS KMS。独自の KMS キーを提供することを選択した場合、ID マッピングテーブルのコンテンツは保管時に を介して KMS キーで暗号化されます AWS Entity Resolution。ID マッピングワークフローで暗号化を使用するために必要なアクセス許可の詳細については、「AWS Entity Resolution ユーザーガイド」の「[AWS Entity Resolutionのワークフロージョブロールを作成する](#)」を参照してください。

転送中の暗号化

AWS Clean Rooms は転送中の暗号化に Transport Layer Security (TLS) を使用します。との通信 AWS Clean Rooms は常に HTTPS 経由で行われるため、データは Amazon S3、Amazon Athena、Snowflake のいずれに保存されているかに関係なく、転送中に常に暗号化されます。これには、Clean Rooms ML を使用するとき転送中のすべてのデータが含まれます。

基になるデータの暗号化

基になるデータを暗号化する方法の詳細については、「[Cryptographic Computing for Clean Rooms](#)」を参照してください。

キーポリシー

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが 1 つだけ必要です。このポリシーには、そのキーを使用できるユーザーとその使用方法を決定するステートメントが含まれています。キーポリシーは、カスタマーマネージドキーの作成時に指定できます。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へのアクセスの管理」を参照してください。

AWS Clean Rooms カスタム ML モデル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:DescribeKey – がキー AWS Clean Rooms を検証できるように、カスタマーマネージドキーの詳細を提供します。
- kms:Decrypt – 暗号化されたデータを復号し、関連するジョブで使用する AWS Clean Rooms ためのへのアクセスを提供します。
- kms:CreateGrant - Clean Rooms ML は、Amazon ECR の許可を作成することで、Amazon ECR の保管中のトレーニングイメージと推論イメージを暗号化します。詳細については、「[Amazon ECR での保管時の暗号化](#)」を参照してください。Clean Rooms ML は、Amazon SageMaker AI を使用してトレーニングジョブと推論ジョブを実行し、SageMaker AI に許可を作成して、インスタンスにアタッチされた Amazon EBS ボリュームと Amazon S3 の出力データを暗号化します。詳細については、[Amazon SageMaker AI で暗号化を使用して保管中のデータを保護する](#)」を参照してください。
- kms:GenerateDataKey - Clean Rooms ML は、サーバー側の暗号化を使用して Amazon S3 に保存されている保管中のデータを暗号化します AWS KMS keys。詳細については、「[Amazon S3 での AWS KMS keys \(SSE-KMS\) によるサーバー側の暗号化の使用Amazon S3](#)」を参照してください。

以下は、次のリソースの AWS Clean Rooms に追加できるポリシーステートメントの例です。

合成データを含む ML 入力チャネル

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access to principals authorized to use AWS Clean Rooms ML",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::666666666666:role/ExampleRole"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "cleanrooms-ml.us-east-1.amazonaws.com"
        },
        "ForAllValues:StringEquals": {
          "kms:GrantOperations": [
            "Decrypt",
            "Encrypt",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "CreateGrant",
            "DescribeKey",
            "RetireGrant",
            "GenerateDataKey"
          ]
        },
        "BoolIfExists": {
          "kms:GrantIsForAWSResource": true
        }
      }
    },
    {
      "Sid": "Allow describe key for principals authorized to use AWS Clean Rooms ML",
```

```

    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
    },
    "Action": [
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "cleanrooms-ml.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "Sid": "Allow grant operations for AWS Clean Rooms ML service principal",
    "Effect": "Allow",
    "Principal": {
      "Service": "cleanrooms-ml.amazonaws.com"
    },
    "Action": [
      "kms:GenerateDataKey",
      "kms:CreateGrant",
      "kms:Decrypt"
    ],
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "kms:GrantOperations": [
          "Decrypt",
          "Encrypt",
          "GenerateDataKeyWithoutPlaintext",
          "ReEncryptFrom",
          "ReEncryptTo",
          "CreateGrant",
          "DescribeKey",
          "RetireGrant",
          "GenerateDataKey"
        ]
      }
    }
  },
  {
    "Sid": "Allow describe key for AWS Clean Rooms ML service principal",

```

```

    "Effect": "Allow",
    "Principal": {
      "Service": "cleanrooms-ml.amazonaws.com"
    },
    "Action": [
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }
]
}

```

合成データのない ML 入力チャネル

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access to principals authorized to use AWS Clean Rooms ML",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::666666666666:role/ExampleRole"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "cleanrooms-ml.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "Allow access to AWS Clean Rooms ML service principal",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms-ml.amazonaws.com"
      },
    }
  ]
}

```

```

    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }
]
}

```

トレーニング済みモデルジョブまたはトレーニング済みモデル推論ジョブ

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow grant operations for principals authorized to use AWS  
Clean Rooms ML",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::666666666666:role/ExampleRole"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "cleanrooms-ml.us-east-1.amazonaws.com"
        },
        "ForAllValues:StringEquals": {
          "kms:GrantOperations": [
            "Decrypt",
            "Encrypt",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "CreateGrant",

```

```

        "DescribeKey",
        "RetireGrant",
        "GenerateDataKey"
    ],
    },
    "BoolIfExists": {
        "kms:GrantIsForAWSResource": true
    }
},
{
    "Sid": "Allow describe key for principals authorized to use AWS Clean
Rooms ML",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
    },
    "Action": [
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "cleanrooms-ml.us-east-1.amazonaws.com"
        }
    }
},
{
    "Sid": "Allow grant operations for AWS Clean Rooms ML service
principal",
    "Effect": "Allow",
    "Principal": {
        "Service": "cleanrooms-ml.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:Decrypt"
    ],
    "Resource": "*",
    "Condition": {
        "ForAllValues:StringEquals": {
            "kms:GrantOperations": [
                "Decrypt",

```

```

        "Encrypt",
        "GenerateDataKeyWithoutPlaintext",
        "ReEncryptFrom",
        "ReEncryptTo",
        "CreateGrant",
        "DescribeKey",
        "RetireGrant",
        "GenerateDataKey"
    ]
}
},
{
    "Sid": "Allow describe key for AWS Clean Rooms ML service principal",
    "Effect": "Allow",
    "Principal": {
        "Service": "cleanrooms-ml.amazonaws.com"
    },
    "Action": [
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
]
}

```

Clean Rooms ML は、カスタマーマネージドキーポリシーでのサービス暗号化コンテキストまたはソースコンテキストの指定をサポートしていません。サービスによって内部的に使用される暗号化コンテキストは、CloudTrail の顧客に表示されます。

のサービスにリンクされたロールの使用 AWS Clean Rooms

AWS Clean Rooms 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 Clean Rooms。サービスにリンクされたロールは、によって事前定義 AWS Clean Rooms 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 Clean Rooms 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 Clean Rooms 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 Clean Rooms することができます。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 Clean Rooms 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」を持つ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 Clean Rooms

AWS Clean Rooms は、AWSServiceRoleForAWSCleanRooms 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、Clean Rooms 関連の CloudWatch メトリクスを AWS アカウントに発行します。

AWSServiceRoleForAWSCleanRooms 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。

- `cleanrooms.amazonaws.com`

AWSCleanRoomsServiceRolePolicy という名前のロールアクセス許可ポリシーにより AWS Clean Rooms、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。

- アクション: `cloudwatch:PutMetricData`。対象リソース: `all AWS resources, restricted to the AWS Clean Rooms namespace`

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 Clean Rooms

IAM コンソールを使用して、AWSServiceRoleForAWSCleanRooms ユースケースでサービスにリンクされたロールを作成できます。AWS CLI または AWS API で、サービス名を使用し

で `cleanrooms.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。

のサービスにリンクされたロールの編集 AWS Clean Rooms

AWS Clean Rooms では、`AWSServiceRoleForAWSCleanRooms` 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 Clean Rooms

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、積極的にモニタリングまたは保守されていない未使用のエンティティを排除できます。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。

`AWSServiceRoleForAWSCleanRooms` を削除するには、まず [のすべてのコラボレーションとメンバーシップ](#) を削除する必要があります AWS アカウント。

Note

リソースを削除しようとしたときに AWS Clean Rooms 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、`AWSServiceRoleForAWSCleanRooms` 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 Clean Rooms 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 Clean Rooms 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。

のデータ保持 AWS Clean Rooms

一時的に AWS Clean Rooms コラボレーションに読み込まれたデータは、クエリの完了後に削除されます。

類似モデルを作成すると、Clean Rooms ML はトレーニングデータを読み取り、ML モデルに適した形式に変換し、トレーニング済みのモデルパラメータを Clean Rooms ML 内に保存します。Clean Rooms ML はトレーニングデータのコピーを保持しません。AWS Clean Rooms SQL クエリは、クエリの実行後にデータを保持しません。次に、Clean Rooms ML はトレーニング済みのモデルを使用して全ユーザーの行動を要約します。Clean Rooms ML は、類似モデルがアクティブである限り、各ユーザーのユーザーレベルのデータセットをデータに保存します。

類似セグメント生成ジョブを開始すると、Clean Rooms ML はシードデータを読み取り、関連する類似モデルから動作概要を読み取り、AWS Clean Rooms サービス内に保存される類似セグメントを作成します。Clean Rooms ML はシードデータのコピーを保持しません。Clean Rooms ML は、ジョブがアクティブである限り、ジョブのユーザーレベルの出力を保存します。

シードデータが SQL クエリから取得された場合、そのクエリの出力はジョブの期間中のみサービスに保存されます。クエリの結果は、保管中および転送中に暗号化されます。

類似モデルまたは類似セグメント生成ジョブデータを削除する場合は、API を使用して削除します。Clean Rooms ML は、モデルまたはジョブに関連するすべてのデータを非同期的に削除します。このプロセスが完了すると、Clean Rooms ML はモデルまたはジョブのメタデータを削除し、API には表示されなくなります。Clean Rooms ML は、ディザスタリカバリ対策のため、削除されたデータを 3 日間保持します。ジョブまたはモデルが API に表示されなくなり、3 日が経過すると、モデルまたはジョブに関連するすべてのデータが完全に削除されます。

でのデータコラボレーションのベストプラクティス AWS Clean Rooms

このトピックでは、AWS Clean Roomsでデータコラボレーションを行うときのベストプラクティスについて説明します。

AWS Clean Rooms は、コラボレーションで機密データを保護する機能を強化するために設定できる [AWS 責任共有 Model](#)。AWS Clean Rooms offers [分析ルール](#)に従います。で設定した分析ルール AWS Clean Rooms は、設定した制限 (クエリコントロールとクエリ出力コントロール) を適用します。制限の決定とそれに応じた分析ルールの設定はユーザーの責任となります。

データコラボレーションには、単に を使用する以上のものが含まれる場合があります AWS Clean Rooms。データコラボレーションの利点を最大化するために、 を使用し、AWS Clean Rooms 特に分析ルールを使用して、次のベストプラクティスを実行することをお勧めします。

トピック

- [を使用したベストプラクティス AWS Clean Rooms](#)
- [で分析ルールを使用するためのベストプラクティス AWS Clean Rooms](#)

を使用したベストプラクティス AWS Clean Rooms

各データコラボレーションのリスクを評価し、それを外部および内部のコンプライアンスプログラムやポリシーなどのプライバシー要件と比較する責任はユーザーにあります。を使用する場合は、追加のアクションを実行することをお勧めします AWS Clean Rooms。これらの措置によって、リスクをさらに管理し、第三者がデータを再特定しようとする試み (差分攻撃やサイドチャネル攻撃など) を防止できる場合があります。

例えば、コラボレーションを開始する前に、相手のコラボレーターに対してデューデリジェンスを実施し、法的契約を結ぶことを検討してください。自社データの使用状況を監視するために、AWS Clean Roomsの使用に際して他の監査メカニズムを採用することも検討してください。

で分析ルールを使用するためのベストプラクティス AWS Clean Rooms

の分析ルール AWS Clean Rooms では、設定されたテーブルにクエリコントロールを設定することで、実行できるクエリを制限できます。例えば、設定済みテーブルを結合する方法や、選択できる列に関するクエリコントロールを設定できます。また、出力行の集約しきい値などのクエリ結果コントロールを設定して、クエリ出力を制限することもできます。クエリに含まれる設定済みテーブルでメ

ンバーが設定した分析ルールを満たしていない行については、クエリが一切拒否され、結果から除外されます。

設定済みテーブルで分析ルールを使用する際には、次の 10 のベストプラクティスに従うことをお勧めします。

- クエリのユースケース (オーディエンスプランニングやアトリビューションなど) ごとに、個別の設定済みテーブルを作成します。同じ AWS Glue テーブルを基に、複数の設定済みテーブルを作成できます。
- コラボレーションでのクエリに必要な列 (ディメンション列、リスト列、結合列など) を分析ルールに指定します。これにより、差分攻撃や、他のメンバーによってデータのリバースエンジニアリングが行われるリスクを軽減できる可能性があります。許可リスト列の機能を使用して、今後クエリを実行できるようにしたい他の列を記録しておきます。特定のコラボレーションに使用できる列をカスタマイズするには、同じ基盤となるテーブルを持つ追加の設定済み AWS Glue テーブルを作成します。
- コラボレーションでの分析に必要な関数を分析ルールに指定します。これにより、個々のデータポイントの情報が提示されるまれな関数エラーによるリスクを軽減できます。特定のコラボレーションに使用できる関数をカスタマイズするには、基になる同じ AWS Glue テーブルを使用して追加の設定済みテーブルを作成します。
- 行レベルの値が機密であるすべての列に集約制約を追加します。これには、設定済みテーブル内の列のうち、コラボレーションの他のメンバーのテーブル、および集約制約として分析ルールにも存在する列が含まれます。また、設定済みテーブル内のクエリを実行できない列、つまり、設定済みテーブルには存在するが分析ルールには存在しない列も含まれます。集約制約により、クエリ結果がコラボレーション外部のデータに関連付けられるリスクを軽減できます。
- テスト用のコラボレーションと分析ルールを作成して、指定された分析ルールで作成した制限をテストします。
- コラボレーターが設定したテーブルと、設定済みテーブルにあるメンバーの分析ルールを確認して、コラボレーションに際して合意された内容と一致していることを確認します。これにより、他のメンバーが独自のデータを設計して、合意されていないクエリを実行するリスクを軽減できます。
- 分析ルールの設定後に、設定済みテーブルで有効になっている、提示されているクエリの例 (コンソールのみ) を確認します。

Note

提示されているクエリの例以外に、分析ルールや、他のコラボレーションメンバーのテーブルと分析ルールに基づいて、その他のクエリも利用できます。

- コラボレーションの設定済みテーブルの分析ルールは、追加または更新が可能です。追加や更新を行う際は、設定済みテーブルが関連付けられているすべてのコラボレーションと、その結果生じる影響を確認してください。これにより、すべてのコラボレーションで古い分析ルールが使用されることを回避できます。
- コラボレーションで実行されるクエリを確認して、そのクエリがコラボレーションに際して合意されたユースケースまたはクエリと一致していることを確認します ([クエリログ記録] 機能がオンになっている場合は、クエリがクエリログに表示されます)。これにより、合意されていない分析をメンバーが実行するリスクや、サイドチャネル攻撃などの潜在的な攻撃のリスクを軽減できます。
- コラボレーションメンバーの分析ルールやクエリで使用する設定済みテーブルの列を確認して、コラボレーションに際して合意された内容と一致していることを確認します (クエリのログ記録機能がオンになっている場合は、クエリがクエリログに表示されます)。これにより、他のメンバーが独自のデータを設計して、合意されていないクエリを実行するリスクを軽減できます。

の Identity and Access Management AWS Clean Rooms

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 Clean Rooms 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、追加料金なしで利用できる 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](#)
- [が IAM と AWS Clean Rooms 連携する方法](#)
- [のアイデンティティベースのポリシーの例 AWS Clean Rooms](#)
- [AWS の 管理ポリシー AWS Clean Rooms](#)
- [AWS Clean Rooms ID とアクセスのトラブルシューティング](#)

- [サービス間の混乱した代理の防止](#)
- [AWS Clean Rooms ML の IAM 動作](#)
- [クリーンルーム ML カスタムモデルの IAM 動作](#)

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 (「[AWS Clean Rooms ID とアクセスのトラブルシューティング](#)」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 (「[が IAM と AWS Clean Rooms 連携する方法](#)」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 (「[のアイデンティティベースのポリシーの例 AWS Clean Rooms](#)」を参照)

アイデンティティを使用した認証

認証は、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして 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によって、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。AWS IAM アイデンティティセンター (IAM Identity Center) ユーザーまたは会社のシングルサインオン認証は、フェデレーティッドアイデンティティの例です。フェデレーティッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、AWS マネジメントコンソール または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「[AWS サインイン ユーザーガイド](#)」の「[へのサインイン AWS アカウント](#)方法」を参照してください。

AWS プログラムで にアクセスする場合、はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストを暗号化して署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用して自分でリクエストに署名する方法の詳細については、「AWS 全般のリファレンス」の「[署名バージョン 4 の署名プロセス](#)」を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。たとえば、AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを強化することをお勧めします。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「[Multi-factor authentication](#)」(多要素認証) および「IAM ユーザーガイド」の「[AWSでの多要素認証 \(MFA\) の使用](#)」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、アカウント内のすべての 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 1 つのサインインアイデンティティから始めます。このアイデンティティは AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることによってアクセスできます。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。ルートユーザーの認証情報を保護し、この認証情報はルートユーザーのみが実行できるタスクに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「AWS 全般のリファレンス」の「[AWS アカウントのルートユーザーの認証情報と IAM ID](#)」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 にアクセスすることを人間のユーザーに要求する 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 AWS、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) 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。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。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構造と内容の詳細については、「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 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。

すべての IAM エンティティ (ユーザーまたはロール) は、許可のない状態からスタートします。デフォルトでは、ユーザーは何もできず、自分のパスワードを変更することすらできません。何かを実行する許可をユーザーに付与するには、管理者がユーザーに許可ポリシーをアタッチする必要があります。また、管理者は、必要な許可があるグループにユーザーを追加できます。管理者がグループに許可を付与すると、そのグループ内のすべてのユーザーにこれらの許可が付与されます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、iam:GetRole アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、AWS マネジメントコンソール、AWS CLI または AWS API からロール情報を取得できます。

アイデンティティベースポリシー

アイデンティティベースのポリシーは、IAM ユーザーグループ、ユーザーのグループ、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。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれます。マネージドポリシーは、AWS アカウント内の複数のユーザー、グループ、およびロールにアタッチできるスタンドアロンポリシーです。管理ポリシーには、AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたは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[マネージドポリシーとインラインポリシーの比較](#)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。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、フェデレーティッドユーザー、またはを含めることができます AWS のサービス。

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、一般的でない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- **アクセス許可の境界** - アクセス許可の境界は、アイデンティティベースポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティに許可の境界を設定できます。結果として許可される範囲は、エンティティのアイデンティティベースポリシーとその許可の境界の共通部分になります。Principal フィールドでユー

ザーまたはロールを指定するリソースベースのポリシーでは、許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可の境界](#)」を参照してください。

- サービスコントロールポリシー (SCPs) – SCPsは、の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。AWS Organizations は、ビジネスが所有する複数の をグループ化して一元管理するためのサービス AWS アカウント です。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「[SCP の仕組み](#)」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。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。

が IAM と AWS Clean Rooms 連携する方法

IAM を使用して へのアクセスを管理する前に AWS Clean Rooms、 で使用できる IAM 機能を確認してください AWS Clean Rooms。

で使用できる IAM 機能 AWS Clean Rooms

IAM 機能	AWS Clean Rooms 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	部分的

IAM 機能	AWS Clean Rooms サポート
ポリシーアクション	あり
ポリシーリソース	はい
ポリシー条件キー (サポート固有)	部分的
ACL	なし
ABAC (ポリシー内のタグ)	あり
一時的な認証情報	あり
転送アクセスセッション (FAS)	あり
サービスロール	あり
サービスリンクロール	いいえ

AWS Clean Rooms およびその他の がほとんどの IAM 機能と AWS のサービス 連携する方法の概要を把握するには、[AWS のサービス「IAM ユーザーガイド」の「IAM と連携する」](#)を参照してください。

のアイデンティティベースのポリシー AWS Clean Rooms

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。

のアイデンティティベースのポリシーの例 AWS Clean Rooms

AWS Clean Rooms アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS Clean Rooms](#)。

内のリソースベースのポリシー AWS Clean Rooms

リソースベースのポリシーのサポート: 一部

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、フェデレーティッドユーザー、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。

この AWS Clean Rooms サービスは、設定された類似モデルマネージドリソースポリシーと呼ばれる 1 つのタイプのリソースベースのポリシーのみをサポートします。これは、設定された類似モデルにアタッチされます。このポリシーは、設定済みの類似モデルでアクションを実行できるプリンシパルを定義します。

リソースベースのポリシーを設定済みの類似モデルにアタッチする方法については、「[AWS Clean Rooms ML の IAM 動作](#)」を参照してください。

のポリシーアクション AWS Clean Rooms

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 Clean Rooms 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「[で定義されるアクション AWS Clean Rooms](#)」を参照してください。

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 AWS Clean Rooms を使用します。

```
cleanrooms
```

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": [  
    "cleanrooms:action1",  
    "cleanrooms:action2"  
]
```

AWS Clean Rooms アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS Clean Rooms](#)。

のポリシーリソース AWS Clean Rooms

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 Clean Rooms リソースタイプとその ARNs 「[で定義されるリソース AWS Clean Rooms](#)」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「[AWS Clean Roomsで定義されるアクション](#)」を参照してください。

AWS Clean Rooms アイデンティティベースのポリシーの例を表示するには、「」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例 AWS Clean Rooms](#)。

のポリシー条件キー AWS Clean Rooms

サービス固有のポリシー条件キーのサポート: 部分的

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 Clean Rooms ML がポリシー条件キーを使用する方法については、「」を参照してください。[AWS Clean Rooms ML の IAM 動作](#)。

ACLs AWS Clean Rooms

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。

を使用した ABAC AWS Clean Rooms

ABAC (ポリシー内のタグ) のサポート: あり

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コントロール \(ABAC\) を使用する](#)」を参照してください。

での一時的な認証情報の使用 AWS Clean Rooms

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イッチロールの使用時に自動的に作成されます。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 AWS 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。

の転送アクセスセッション AWS Clean Rooms

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。

のサービスロール AWS Clean Rooms

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 Clean Rooms 機能が破損する可能性があります。AWS Clean Rooms が指示する場合にのみ、サービスロールを編集します。

のサービスにリンクされたロール AWS Clean Rooms

サービスにリンクされたロールのサポート: なし

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、サービスにリンクされた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、「[IAM と提携するAWS のサービス](#)」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされた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、[はい] リンクを選択します。

のアイデンティティベースのポリシーの例 AWS Clean Rooms

デフォルトでは、ユーザーおよびロールには、AWS Clean Rooms リソースを作成または変更する権限はありません。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 Clean Rooms、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「[のアクション、リソース、および条件キー AWS Clean Rooms](#)」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 Clean Rooms 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](#)

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 Clean Rooms リソースを誰かが作成、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

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 使用できます 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 Clean Rooms コンソールの使用

AWS Clean Rooms 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 Clean Rooms 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。

ユーザーとロールが AWS Clean Rooms 引き続きコンソールを使用できるようにするには、エンティティに AWS Clean Rooms *FullAccess* または *ReadOnly* AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の「[ユーザーへのアクセス許可の追加](#)」を参照してください。

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": "*"
}
]
}
```

AWS の 管理ポリシー AWS Clean Rooms

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、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るように、多くの一般的なユースケースにアクセス許可を提供するように設計されています。

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ります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用でき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 管理ポリシー: **AWSCleanRoomsReadOnlyAccess**

AWSCleanRoomsReadOnlyAccess を IAM プリンシパルにアタッチできます。

このポリシーは、AWSCleanRoomsReadOnlyAccess コラボレーションのリソースとメタデータへの読み取り専用のアクセス許可を付与します。

アクセス許可の詳細

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。

- `CleanRoomsRead` - プリンシパルにサービスへの読み取り専用アクセスを許可します。
- `ConsoleDisplayTables` - プリンシパルに、基盤となる AWS Glue テーブルに関するデータをコンソールに表示するために必要な AWS Glue メタデータへの読み取り専用アクセスを許可します。
- `ConsoleLogSummaryQueryLogs` - プリンシパルにクエリログの閲覧を許可します。
- `ConsoleLogSummaryObtainLogs` - プリンシパルにログ結果の取得を許可します。

ポリシーの詳細の JSON リストについては、AWS 「マネージドポリシーリファレンスガイド」の[AWSCleanRoomsReadOnlyAccess](#)」を参照してください。

AWS 管理ポリシー: **AWSCleanRoomsFullAccess**

`AWSCleanRoomsFullAccess` を IAM プリンシパルにアタッチできます。

このポリシーは、AWS Clean Rooms コラボレーションのリソースとメタデータへのフルアクセス (読み取り、書き込み、更新) を許可する管理アクセス許可を付与します。このポリシーには、クエリを実行するためのアクセス許可が含まれています。

アクセス許可の詳細

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。

- `CleanRoomsAccess` - すべてのリソースに対するすべてのアクションへのフルアクセスを許可します AWS Clean Rooms。
- `PassServiceRole` - 名前に「cleanrooms」が含まれるサービスのみにサービスロールを渡すためのアクセス許可 (`PassedToService` 条件) を付与します。
- `ListRolesToPickServiceRole` - プリンシパルが使用時にサービスロールを選択できるように、すべてのロールを一覧表示できるようにします AWS Clean Rooms。
- `GetRoleAndListRolePoliciesToInspectServiceRole` - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- `ListPoliciesToInspectServiceRolePolicy` - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- `GetPolicyToInspectServiceRolePolicy` - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。

- `ConsoleDisplayTables` – プリンシパルに、基盤となる AWS Glue テーブルに関するデータをコンソールに表示するために必要な AWS Glue メタデータへの読み取り専用アクセスを許可します。
- `ConsolePickQueryResultsBucketListAll` – 使用可能なすべての Amazon S3 バケットのリストから、クエリ結果を書き込む S3 バケットを選択することをプリンシパルに許可します。
- `SetQueryResultsBucket` – クエリ結果を書き込む S3 バケットを選択することをプリンシパルに許可します。
- `ConsoleDisplayQueryResults` – S3 バケットから読み取ったクエリ結果を顧客に示すことをプリンシパルに許可します。
- `WriteQueryResults` – クエリ結果を顧客所有の S3 バケットに書き込むことをプリンシパルに許可します。
- `EstablishLogDeliveries` – 顧客の Amazon CloudWatch Logs ロググループにクエリログを提供することをプリンシパルに許可します。
- `SetupLogGroupsDescribe` – Amazon CloudWatch Logs ロググループの作成プロセスを使用することをプリンシパルに許可します。
- `SetupLogGroupsCreate` – プリンシパルに Amazon CloudWatch Logs ロググループの作成を許可します。
- `SetupLogGroupsResourcePolicy` – Amazon CloudWatch Logs ロググループにリソースポリシーを設定することをプリンシパルに許可します。
- `ConsoleLogSummaryQueryLogs` – プリンシパルにクエリログの閲覧を許可します。
- `ConsoleLogSummaryObtainLogs` – プリンシパルにログ結果の取得を許可します。

ポリシーの詳細の JSON リストについては、AWS 「マネージドポリシーリファレンスガイド」の[AWSCleanRoomsFullAccess](#)」を参照してください。

AWS 管理ポリシー: **AWSCleanRoomsFullAccessNoQuerying**

IAM principals に `AWSCleanRoomsFullAccessNoQuerying` をアタッチできます。

このポリシーは、AWS Clean Rooms コラボレーションのリソースとメタデータへのフルアクセス (読み取り、書き込み、更新) を許可する管理アクセス許可を付与します。このポリシーでは、クエリを実行するためのアクセス許可は除外されます。

アクセス許可の詳細

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。

- `CleanRoomsAccess` – コラボレーションでのクエリを除く AWS Clean Rooms、 のすべてのリソースに対するすべてのアクションへのフルアクセスを許可します。
- `CleanRoomsNoQuerying` – `StartProtectedQuery` と `UpdateProtectedQuery` を明示的に拒否し、クエリを禁止します。
- `PassServiceRole` - 名前に「cleanrooms」が含まれるサービスのみにサービスロールを渡すためのアクセス許可 (`PassedToService` 条件) を付与します。
- `ListRolesToPickServiceRole` – プリンシパルが使用時にサービスロールを選択できるように、すべてのロールを一覧表示できるようにします AWS Clean Rooms。
- `GetRoleAndListRolePoliciesToInspectServiceRole` - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- `ListPoliciesToInspectServiceRolePolicy` - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- `GetPolicyToInspectServiceRolePolicy` - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- `ConsoleDisplayTables` – プリンシパルに、基盤となる AWS Glue テーブルに関するデータをコンソールに表示するために必要な AWS Glue メタデータへの読み取り専用アクセスを許可します。
- `EstablishLogDeliveries` – 顧客の Amazon CloudWatch Logs ロググループにクエリログを提供することをプリンシパルに許可します。
- `SetupLogGroupsDescribe` – Amazon CloudWatch Logs ロググループの作成プロセスを使用することをプリンシパルに許可します。
- `SetupLogGroupsCreate` – プリンシパルに Amazon CloudWatch Logs ロググループの作成を許可します。
- `SetupLogGroupsResourcePolicy` – Amazon CloudWatch Logs ロググループにリソースポリシーを設定することをプリンシパルに許可します。
- `ConsoleLogSummaryQueryLogs` - プリンシパルにクエリログの閲覧を許可します。
- `ConsoleLogSummaryObtainLogs` - プリンシパルにログ結果の取得を許可します。
- `cleanrooms` – AWS Clean Rooms サービス内のコラボレーション、分析テンプレート、設定済みテーブル、メンバーシップ、および関連リソースを管理します。これらのリソースに関する情報の作成、更新、削除、一覧表示、取得など、さまざまな操作を実行します。
- `iam-cleanrooms` 「」を含む名前のサービスロールを AWS Clean Rooms サービスに渡します。ロール、ポリシーを一覧表示し、サービスに関連する AWS Clean Rooms サービスロールとポリシーを検査します。

- `glue` – データベース、テーブル、パーティション、スキーマに関する情報を から取得します AWS Glue。これは、AWS Clean Rooms サービスが基盤となるデータソースを表示して操作するために必要です。
- `logs` – CloudWatch Logs のログ配信、ロググループ、およびリソースポリシーを管理します。AWS Clean Rooms サービスに関連するログをクエリして取得します。これらのアクセス許可は、サービス内のモニタリング、監査、およびトラブルシューティングの目的で必要になります。

また、ポリシーは、`cleanrooms:StartProtectedQuery` および `cleanrooms:UpdateProtectedQuery` アクションを明示的に拒否し、ユーザーが保護されたクエリを直接実行または更新できないようにします。これは、AWS Clean Rooms により制御されたメカニズムを介して実行する必要があります。

ポリシーの詳細の JSON リストについては、AWS 「マネージドポリシーリファレンスガイド」の[AWSCleanRoomsFullAccessNoQuerying](#)」を参照してください。

AWS 管理ポリシー: **AWSCleanRoomsMLReadOnlyAccess**

`AWSCleanRoomsMLReadOnlyAccess` を IAM プリンシパルにアタッチできます。

このポリシーは、`AWSCleanRoomsMLReadOnlyAccess` コラボレーションのリソースとメタデータへの読み取り専用のアクセス許可を付与します。

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。

- `CleanRoomsConsoleNavigation` – AWS Clean Rooms コンソールの画面を表示するアクセス許可を付与します。
- `CleanRoomsMLRead` – プリンシパルに Clean Rooms ML サービスへの読み取り専用アクセスを許可します。
- `PassCleanRoomsResources` – 指定された AWS Clean Rooms リソースを渡すためのアクセス許可を付与します。

ポリシーの詳細の JSON リストについては、AWS 「マネージドポリシーリファレンスガイド」の[AWSCleanRoomsMLReadOnlyAccess](#)」を参照してください。

AWS 管理ポリシー: **AWSCleanRoomsMLFullAccess**

AWSCleanRoomsMLFullAccess を IAM プリンシパルにアタッチできます。このポリシーは、Clean Rooms ML が必要とするリソースとメタデータへの完全なアクセス (読み取り、書き込み、および更新) を可能にする管理アクセス許可を付与します。

アクセス許可の詳細

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。

- **CleanRoomsMLFullAccess** – すべての Clean Rooms ML アクションへのアクセスを許可します。
- **PassServiceRole** - 名前に「cleanrooms-ml」が含まれるサービスのみにサービスロールを渡すためのアクセス許可 (PassedToService 条件) を付与します。
- **CleanRoomsConsoleNavigation** – AWS Clean Rooms コンソールの画面を表示するアクセス許可を付与します。
- **CollaborationMembershipCheck** – コラボレーション内でオーディエンス生成 (類似セグメント) ジョブを開始すると、Clean Rooms ML サービスは ListMembers を呼び出して、コラボレーションが有効であること、発信者がアクティブなメンバーであること、設定されたオーディエンスモデル所有者がアクティブなメンバーであることを確認します。このアクセス許可は常に必要です。コンソールナビゲーション SID はコンソールユーザーにのみ必要です。
- **PassCleanRoomsResources** – 指定された AWS Clean Rooms リソースを渡すためのアクセス許可を付与します。
- **AssociateModels** – Clean Rooms ML モデルをコラボレーションに関連付けることをプリンシパルに許可します。
- **TagAssociations** – 類似モデルとコラボレーションの関連付けにタグを追加することをプリンシパルに許可します。
- **ListRolesToPickServiceRole** – プリンシパルが使用時にサービスロールを選択できるように、すべてのロールを一覧表示できるようにします AWS Clean Rooms。
- **GetRoleAndListRolePoliciesToInspectServiceRole** - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- **ListPoliciesToInspectServiceRolePolicy** - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。
- **GetPolicyToInspectServiceRolePolicy** - IAM のサービスロールと対応するポリシーを表示することをプリンシパルに許可します。

- `ConsoleDisplayTables` – プリンシパルに、基盤となる AWS Glue テーブルに関するデータをコンソールに表示するために必要な AWS Glue メタデータへの読み取り専用アクセスを許可します。
- `ConsolePickOutputBucket` – 設定済みのオーディエンスモデル出力用の Amazon S3 バケットを選択することをプリンシパルに許可します。
- `ConsolePickS3Location` – 設定済みのオーディエンスモデル出力のバケット内の場所を選択することをプリンシパルに許可します。
- `ConsoleDescribeECRRepositories` – プリンシパルが Amazon ECR リポジトリとイメージを記述できるようにします。

ポリシーの詳細の JSON リストについては、AWS 「マネージドポリシーリファレンスガイド」の [AWSCleanRoomsMLFullAccess](#)」を参照してください。

AWS Clean Rooms AWS 管理ポリシーの更新

このサービスがこれらの変更の追跡を開始 AWS Clean Rooms してからの の AWS 管理ポリシーの更新に関する詳細を表示します。このページの変更に関する自動アラートについては、AWS Clean Rooms ドキュメント履歴ページの RSS フィードにサブスクライブしてください。

変更	説明	日付
AWSCleanRoomsFullAccessNoQuering – 既存のポリシーの更新	クリーンルーム:UpdateConfiguredTableAllowedColumns とクリーンルーム:UpdateConfiguredTableReference を追加しましたCleanRoomsAccess。	2025 年 7 月 29 日
AWSCleanRoomsMLReadOnlyAccess – 既存のポリシーの更新	PassCleanRoomsResources が AWSCleanRoomsMLReadOnlyAccess に追加されました。PassCleanRoomsResources および ConsoleDescribeECRRepositories が AWSCleanRoomsMLFullAccess に追加されました。	2025 年 1 月 10 日
AWSCleanRoomsFullAccessNoQuering – 既存のポリシーの更新	cleanrooms:BatchGetSchemaAnalysisRule が CleanRoomsAccess に追加されました。	2024 年 5 月 13 日

変更	説明	日付
AWSCleanRoomsFullAccess – 既存のポリシーの更新	このポリシーの AWSCleanRoomsFullAccess のステートメント ID を ConsolePickQueryResultsBucket から SetQueryResultsBucket に更新し、アクセス許可をより適切に表現しました。これは、コンソールの有無にかかわらず、クエリ結果バケットの設定にはアクセス許可が必要であるためです。	2024 年 3 月 21 日
AWSCleanRoomsMLReadOnlyAccess - 新しいポリシー AWSCleanRoomsMLFullAccess - 新しいポリシー	AWS Clean Rooms ML をサポートするために AWSCleanRoomsMLReadOnlyAccess とを追加AWSCleanRoomsMLFullAccessしました。	2023 年 11 月 29 日
AWSCleanRoomsFullAccessNoQuering – 既存のポリシーの更新	cleanrooms:CreateAnalysisTemplate、cleanrooms:GetAnalysisTemplate、cleanrooms:UpdateAnalysisTemplate、cleanrooms:DeleteAnalysisTemplate、cleanrooms>ListAnalysisTemplates、cleanrooms:GetCollaborationAnalysisTemplate、cleanrooms:BatchGetCollaborationAnalysisTemplate、および cleanrooms>ListCollaborationAnalysisTemplates が CleanRoomsAccess に追加され、新しい分析テンプレート機能が有効になりました。	2023 年 7 月 31 日

変更	説明	日付
AWSCleanRoomsFullAccessNoQuering – 既存のポリシーの更新	cleanrooms:ListTagsForResource、cleanrooms:UntagResource、および cleanrooms:TagResource が CleanRoomsAccess に追加され、リソースのタグ付けが可能になりました。	2023 年 3 月 21 日
AWS Clean Rooms が変更の追跡を開始しました	AWS Clean Rooms は、AWS 管理ポリシーの変更の追跡を開始しました。	2023 年 1 月 12 日

AWS Clean Rooms ID とアクセスのトラブルシューティング

次の情報は、および 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 AWS Clean Rooms と修復に役立ちます。

トピック

- [でアクションを実行する権限がありません AWS Clean Rooms](#)
- [iam:PassRole を実行する権限がありません](#)
- [自分の 以外のユーザーに自分の AWS Clean Rooms リソース AWS アカウント へのアクセスを許可したい](#)

でアクションを実行する権限がありません AWS Clean Rooms

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。

以下のエラー例は、mateojackson IAM ユーザーがコンソールを使用して架空の *my-example-widget* リソースに関する詳細情報を表示しようとしているが、架空の cleanrooms:*GetWidget* 権限がないという場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
cleanrooms:GetWidget on resource: my-example-widget
```

この場合、Mateo のポリシーでは、*my-example-widget* アクションを使用して `cleanrooms:GetWidget` リソースにアクセスすることを許可するように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 Clean Rooms 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、既存のロールをそのサービスに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権限が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して AWS Clean Rooms でアクションを実行しようとする場合に発生します。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権限が必要です。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 以外のユーザーに自分の AWS Clean Rooms 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。

詳細については、以下を参照してください。

- がこれらの機能 AWS Clean Rooms 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 [が IAM と AWS Clean Rooms 連携する方法](#)。

- 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でのロールとリソースベースのポリシーの使用の違いの詳細については、IAM ユーザーガイドの[IAM ロールとリソースベースのポリシーとの相違点](#)を参照してください。

サービス間の混乱した代理の防止

混乱した代理問題は、アクションを実行する許可を持たないエンティティが、より特権のあるエンティティにアクションを実行するように強制できるセキュリティの問題です。では AWS、サービス間のなりすましにより、混乱した代理問題が発生する可能性があります。サービス間でのなりすましは、1 つのサービス (呼び出し元サービス) が、別のサービス (呼び出し対象サービス) を呼び出すときに発生する可能性があります。呼び出し元サービスは、本来ならアクセスすることが許可されるべきではない方法でその許可を使用して、別のお客様のリソースに対する処理を実行するように操作される場合があります。これを防ぐため、AWS では、アカウントのリソースへのアクセス権が付与されたサービスプリンシパルで、すべてのサービスのデータを保護するために役立つツールを提供しています。

リソースポリシーで [aws:SourceArn](#) のグローバル条件コンテキストキーを使用して、AWSClean Rooms が別のサービスに付与するアクセス許可をそのリソースに制限することをお勧めします。クロスサービスのアクセスにリソースを 1 つだけ関連付けたい場合は、aws:SourceArn を使用します。

混乱した代理問題から保護するための最も効果的な方法は、リソースの完全な ARN を指定して aws:SourceArn グローバル条件コンテキストキーを使用することです。では AWS Clean Rooms、sts:ExternalId 条件キーと比較する必要があります。

aws:SourceArn の値は、引き受けられたロールのメンバーシップの ARN に設定する必要があります。

次の例では、AWSClean Rooms で `aws:SourceArn` グローバル条件コンテキストキーを使用して、混乱した代理問題を回避する方法を示します。

 Note

サンプルポリシーは、AWS Clean Rooms が設定済みテーブルのデータとメタデータにアクセスするために使用するサービスロールの信頼ポリシーに適用されます。

`<query-runner-membership-id>` の値は、クエリランナーのメンバーシップ ID に設定する必要があります。

コラボレーションのすべてのメンバーは、設定されたテーブルメタデータを表示できるため、各メンバーシップ ARN をメンバーシップ ARNs のリストに含める必要があります。

 Note

AWS Clean Rooms コンソールを使用してサービスロールを作成すると、コラボレーションの現在のメンバーはすべて、デフォルトで混乱した代理条件に含まれます。

既にテーブルが関連付けられているコラボレーションに新しいメンバーを追加する場合は、サービスロールの混乱した代理条件を新しいメンバーのメンバーシップ ARN で更新してください。

新しいメンバーを追加した後にサービスロールの混乱した代理条件を更新しない場合、その新しいメンバー AWS Clean Rooms はそのロールを使用して取得された の情報にアクセスできなくなります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIfExternalIdMatches",
      "Effect": "Allow",
      "Principal": {
        "Service": "cleanrooms.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
```

```

        "StringLike": {
            "sts:ExternalId": "arn:aws:*:us-east-1:*:dbuser:*/<query-
runner-membership-id>*"
        }
    },
    {
        "Sid": "AllowIfSourceArnMatches",
        "Effect": "Allow",
        "Principal": {
            "Service": "cleanrooms.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "ForAnyValue:ArnEquals": {
                "aws:SourceArn": [
                    "arn:aws:cleanrooms:us-
east-1:111122223333:membership/<member-1-membership-id>",
                    "arn:aws:cleanrooms:us-
east-1:444455556666:membership/<member-2-membership-id>"
                ]
            }
        }
    }
]
}

```

AWS Clean Rooms ML の IAM 動作

クロスアカウントジョブ

Clean Rooms ML を使用すると、ある によって作成された特定のリソース AWS アカウ
ント に別の によってアカウントで安全にアクセスできます AWS アカウ
ント A のクライアントが AWS アカウ
ント B が所有するConfiguredAudienceModelリソー
スStartAudienceGenerationJobで を呼び出すと、Clean Rooms ML はジョブに 2 つの ARNs
を作成します。A の ARN AWS アカウ
ントと AWS アカウ
ント B の ARN。ARNsは、ARN を除いて
同一です AWS アカウ
ント。

Clean Rooms ML は、両方のアカウントがそれぞれ独自の IAM ポリシーをジョブに適用できるよう
に、ジョブ用に 2 つの ARN を作成します。たとえば、両方のアカウントでタグベースのアクセスコ
ントロールを使用し、AWS 組織からのポリシーを適用できます。ジョブは両方のアカウントのデー

タを処理するため、どちらのアカウントでもジョブとそれに関連するデータを削除できます。どちらのアカウントも、もう一方のアカウントによるジョブの削除をブロックすることはできません。

ジョブの実行は 1 つだけで、どちらのアカウントも `ListAudienceGenerationJobs` を呼び出してジョブを確認できます。どちらのアカウントも `Get`、独自の AWS アカウント ID を持つ ARN を使用して、ジョブで、`Delete`、および `Export APIs` を呼び出すことができます。

他の AWS アカウント ID で ARN を使用する場合は、ジョブにアクセス AWS アカウント できません。

ジョブの名前は AWS アカウント内で一意である必要があります。AWS アカウント B の名前は `$accountA-$name` です。AWS アカウント A によって選択された名前は、ジョブが AWS アカウント B で表示されるときに A という AWS アカウント プレフィックスが付けられます。

クロスアカウント `StartAudienceGenerationJob` を成功させるには、AWS アカウント B で次の例のようなリソースポリシーを使用して、AWS アカウント B の新しいジョブと AWS アカウント B `ConfiguredAudienceModel` の新しいジョブの両方でそのアクションを許可する必要があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Clean-Rooms-CAMA-ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "111122223333"
        ]
      },
      "Action": [
        "cleanrooms-ml:StartAudienceGenerationJob"
      ],
      "Resource": [
        "arn:aws:cleanrooms-ml:us-east-1:444455556666:configured-audience-model/id",
        "arn:aws:cleanrooms-ml:us-east-1:444455556666:audience-generation-job/*"
      ],
      "Condition": { "StringEquals": { "cleanrooms-ml:CollaborationId": "UUID" } }
    }
  ]
}
```

```
]
}
```

Note

この AWS Clean Rooms ML リソースポリシーは、クロスアカウントオーディエンス生成をサポートするために 2 つの異なる AWS アカウント IDs を参照します。

- 111122223333 - これは、オーディエンス生成ジョブを開始する権限を持つプリンシパル (ユーザー、ロール、またはサービス) を含むアカウントです。このアカウントは ML 処理ワークフローを開始します。
- 444455556666 - これは ML AWS Clean Rooms リソース (設定されたオーディエンスモデルとオーディエンス生成ジョブ) を所有するアカウントです。このアカウントは ML モデルをホストし、ジョブ実行を管理します。

その他の設定に関する注意事項:

- ステートメント ID (Sid): を実際の AWS Clean Rooms オーディエンスモデルアプリケーション (CAMA) 識別子CAMA-IDに置き換えて、ポリシーステートメントを簡単に識別できるようにします。
- リソース IDs: **ID** を設定したオーディエンスモデルの実際の ID に置き換え、**UUID** を特定のコラボレーション ID に置き換えます。
- 条件: `cleanrooms-ml:CollaborationId` 条件は、指定された AWS Clean Rooms コラボレーションのコンテキスト内でのみオーディエンス生成ジョブを開始できるようにし、追加のセキュリティ境界を提供します。

このクロスアカウント設定により、1 つの組織が ML モデルとインフラストラクチャを管理し、承認されたパートナーがコラボレーション契約の範囲内でオーディエンス生成プロセスを開始できるようにするシナリオが可能になります。

[AWS Clean Rooms ML API](#) を使用して、`true manageResourcePolicies` に設定された類似モデルを作成する場合、はこのポリシー AWS Clean Rooms を作成します。

さらに、AWS アカウント A の発信者の ID ポリシーには、に対する `StartAudienceGenerationJob` アクセス許可が必要です `arn:aws:cleanrooms-ml:us-`

west-1:AccountA:audience-generation-job/*。したがって、A ジョブ、AWS アカウント B StartAudienceGenerationJob AWS アカウント ジョブ、AWS アカウント B アクションの 3 つの IAM リソースがありますConfiguredAudienceModel。

Warning

ジョブ AWS アカウント を開始した は、ジョブに関する AWS CloudTrail 監査ロギイベントを受け取ります。ConfiguredAudienceModel を所有する AWS CloudTrail は AWS アカウント 監査ロギイベントを受信しません。

ジョブのタグ付け

CreateConfiguredAudienceModel の

childResourceTagOnCreatePolicy=FROM_PARENT_RESOURCE パラメータを設定すると、設定した類似モデルから作成されたアカウント内のすべての類似セグメント生成ジョブには、設定した類似モデルと同じタグがデフォルトで割り当てられます。設定した類似モデルが親で、類似セグメント生成ジョブが子です。

自身のアカウント内でジョブを作成する場合、ジョブのリクエストタグは親タグよりも優先されます。他のアカウントが作成したジョブが、自身のアカウントにタグを作成することはありません。childResourceTagOnCreatePolicy=FROM_PARENT_RESOURCE を設定し、別のアカウントでジョブを作成した場合、そのジョブのコピーは 2 つあります。アカウントのコピーには親リソースタグが付けられ、ジョブ送信者のアカウントのコピーにはリクエストのタグが付けられます。

共同作業者の検証

AWS Clean Rooms コラボレーションの他のメンバーにアクセス許可を付与する場合、リソースポリシーには条件キー を含める必要がありますcleanrooms-ml:CollaborationId。これにより、[StartAudienceGenerationJob](#) リクエストに collaborationId パラメータが含まれるように強制されます。collaborationId パラメータがリクエストに含まれると、Clean Rooms ML はコラボレーションが存在すること、ジョブの送信者がコラボレーションのアクティブメンバーであること、設定済みの類似モデルの所有者がコラボレーションのアクティブメンバーであることを検証します。

が設定された類似モデルリソースポリシー AWS Clean Rooms を管理する場合 (`manageResourcePolicies` パラメータは [CreateConfiguredAudienceModelAssociation リクエスト](#) TRUE にあります)、この条件キーはリソースポリシーで設定されます。そのため、[AudienceGenerationJob](#) で collaborationId を指定する必要があります。

クロスアカウントアクセス

アカウント間で呼び出せるのは `StartAudienceGenerationJob` のみです。その他の Clean Rooms ML API はすべて、自身のアカウントのリソースでのみ使用できます。これにより、トレーニングデータ、類似モデルの設定、その他の情報は非公開のままになります。

Clean Rooms ML は、アカウント間で Amazon S3 または AWS Glue ロケーションを公開しません。トレーニングデータの場所、設定済みの類似モデルの出力場所、および類似セグメント生成ジョブシードの場所は、どのアカウントでも表示されません。コラボレーションでクエリログ記録が有効になっていない限り、シードデータが SQL クエリから取得されるかどうかに関わらず、クエリ自体はアカウント間で表示されません。別のアカウントが送信したオーディエンス生成ジョブを Get した場合、サービスにはシードロケーションは表示されません。

クリーンルーム ML カスタムモデルの IAM 動作

クロスアカウントジョブ

Clean Rooms ML では、あるユーザーが作成したコラボレーションに関連付けられた特定のリソース AWS アカウントに、別のユーザーが自分のアカウントで安全にアクセスできます AWS アカウント。クエリを実行するメンバー権限を持つ AWS アカウント A の `ClientConfiguredModelAlgorithmAssociation` は `CreateTrainedModel`、で作成されたカスタム分析ルールで が許可されている場合、コラボレーション内の別のメンバーが所有する `ConfiguredModelAlgorithmAssociation` リソース `StartTrainedModelInferenceJob` で `CreateMLInputChannel`、または を呼び出すことができます `CreateConfiguredTableAnalysisRule`。

さらに、コラボレーションのアクティブなメンバーは、 `DeleteTrainedModelOutput` および `DeleteMLInputChannelData` APIs を介して、トレーニングされたモデルまたは ML 入力チャネルに関連付けられたデータを削除できます。

クロスアカウントアクセス

Clean Rooms ML を使用すると、ユーザーは `GetCollaboration` および `ListCollaboration` APIs。Clean Rooms ML は、KMS キー ARNs、タグ、環境変数、またはハイパーパラメータ (`TrainedModel` アクション用) を他のアカウントに公開しません。

メンバーシップとコラボレーションへのアクセス

Clean Rooms ML カスタムモデルのコンテキストでメンバーシップおよびコラボレーションリソースにアクセスする場合、ユーザーの ID ポリシーにはアクション

`cleanrooms:PassMembership`、`cleanrooms:PassCollaboration`、またはその両方に対するアクセス許可が必要です。を受け入れるすべての APIs には `cleanrooms:PassMembership` 許可 `membershipId` が必要で、を受け入れるすべての APIs には `cleanrooms:PassCollaboration` 許可 `collaborationId` が必要です。コラボレーション ID のコンテキスト `createTrainedModel` で を呼び出すことができるメンバーシップ ID のコンテキスト `GetCollaborationTrainedModel` で を呼び出すことができるロールのサンプル ID ポリシーが提供されます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCleanroomsMLActions",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:PassCollaboration",
        "cleanrooms:PassMembership"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowMembershipAccess",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:GetMembership"
      ],
      "Resource": [
        "arn:aws:cleanrooms:us-east-1:111122223333:membership/memberId"
      ]
    },
    {
      "Sid": "AllowCollaborationAccess",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:GetCollaboration"
      ],
      "Resource": [
```

```
"arn:aws:cleanrooms:us-east-1:111122223333:collaboration/collaborationId"
    ]
  }
]
}
```

のコンプライアンス検証 AWS Clean Rooms

AWS のサービス が特定のコンプライアンスプログラムの対象で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 プログラムによる対象範囲内](#)」の「コンプライアンス」を参照し、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。

の耐障害性 AWS Clean Rooms

AWS グローバルインフラストラクチャは、AWS リージョンとアベイラビリティゾーンを中心に構築されています。リージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立および隔離されたアベイラビリティゾーンがあります。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イ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 リージョンと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。

のインフラストラクチャセキュリティ AWS Clean Rooms

マネージドサービスである AWS Clean Rooms は、AWS グローバルネットワークセキュリティで保護されています。AWS セキュリティサービスと インフラストラクチャ AWS を保護する方法については、[AWS 「クラウドセキュリティ」](#) を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して環境を AWS 設計するには、「Security Pillar AWS Well-Architected Framework」の「[Infrastructure Protection](#)」を参照してください。

AWS が発行した API コールを使用して、ネットワーク AWS Clean Rooms 経由で にアクセスします。クライアントは次をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは Java 7 以降など、ほとんどの最新システムでサポートされています。

ネットワークセキュリティ

クエリの実行中に が S3 バケットから AWS Clean Rooms 読み取ると、 と Amazon S3 間の AWS Clean Rooms トラフィックは AWS プライベートネットワークを介して安全にルーティングされます。処理中のトラフィックは Amazon Signature Version 4 プロトコル (SIGv4) を使用して署名され、HTTPS を使用して暗号化されます。このトラフィックは、設定済みテーブルに設定した IAM サービスロールに基づいて承認されます。

エンドポイント AWS Clean Rooms を介してプログラムで に接続できます。サービスエンドポイントのリストについては、「AWS 全般のリファレンス」の「[AWS Clean Rooms のエンドポイントとクォータ](#)」を参照してください。

すべてのサービスエンドポイントは HTTPS 限定です。VPC AWS Clean Rooms から に接続し、インターネット接続を希望しない場合は、Amazon Virtual Private Cloud (VPC) エンドポイントを使用できます。詳細については、「AWS PrivateLink ガイド」の「[を通じて AWS サービスにアクセスする AWS PrivateLink](#)」を参照してください。

[aws:SourceVpce コンテキストキー](#) を使用する IAM プリンシパルに IAM ポリシーを割り当てて、インターネットではなく VPC エンドポイント AWS Clean Rooms を介してのみ を呼び出すことができるように IAM プリンシパルを制限できます。

インターフェイスエンドポイント (AWS PrivateLink) AWS Clean Rooms を使用して AWS Clean Rooms または ML にアクセスする

AWS PrivateLink を使用して、仮想プライベートクラウド (VPC) と AWS Clean Rooms または ML AWS Clean Rooms の間にプライベート接続を作成できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、AWS Clean Rooms または Direct Connect 接続を使用せずに、VPC 内にあるかのように または AWS Clean Rooms ML にアクセスできます。VPC 内のインスタンスは AWS Clean Rooms にアクセスするためにパブリック IP アドレスを必要としません。

このプライベート接続を確立するには、AWS PrivateLink を利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、AWS Clean Rooms 宛てのトラフィックのエントリポイントとして機能するリクエスト管理型ネットワークインターフェイスです。

詳細については「AWS PrivateLink ガイド」の「[Access AWS のサービス through AWS PrivateLink](#)」を参照してください。

に関する考慮事項 AWS Clean Rooms

のインターフェイスエンドポイントを設定する前に AWS Clean Rooms、「AWS PrivateLink ガイド」の「[考慮事項](#)」を参照してください。

AWS Clean Rooms および AWS Clean Rooms ML 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。

VPC エンドポイントポリシーは、AWS Clean Rooms または ML AWS Clean Rooms ではサポートされていません。デフォルトでは、インターフェイスエンドポイントを介して AWS Clean Rooms および AWS Clean Rooms ML へのフルアクセスが許可されます。または、セキュリティグループをエンドポイントネットワークインターフェイスに関連付けて、インターフェイスエンドポイントを介して AWS Clean Rooms または AWS Clean Rooms ML へのトラフィックを制御することもできます。

のインターフェイスエンドポイントを作成する AWS Clean Rooms

Amazon VPC コンソール AWS Clean Rooms または AWS Command Line Interface () を使用して、または AWS Clean Rooms ML のインターフェイスエンドポイントを作成できます AWS CLI。詳細については、「AWS PrivateLink ガイド」の「[インターフェイスエンドポイントを作成](#)」を参照してください。

次のサービス名 AWS Clean Rooms を使用して のインターフェイスエンドポイントを作成します。

```
com.amazonaws.region.cleanrooms
```

次のサービス名を使用して、ML AWS Clean Rooms のインターフェイスエンドポイントを作成します。

```
com.amazonaws.region.cleanrooms-ml
```

インターフェイスエンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名を使用して、AWS Clean Rooms への API リクエストを実行できます。例えば、`cleanrooms-ml.us-east-1.amazonaws.com`。

モニタリング AWS Clean Rooms

モニタリングは、およびその他の AWS Clean Rooms 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。AWS には、監視 AWS Clean Rooms、問題発生時の報告、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が用意されています。

- Amazon CloudWatch Logs を使用すると、Amazon EC2 インスタンスやその他のソースからログファイルをモニタリング AWS CloudTrail、保存、アクセスできます。Amazon 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できます。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。

Clean Rooms ML により、特定の API アクションのクロスアカウントジョブが許可されます。ジョブ AWS アカウント を開始した は、ジョブの AWS CloudTrail 監査ログイベントを受け取ります。詳細については、[AWS Clean Rooms ML の IAM 動作](#)を参照してください。

- AWS CloudTrail は、によって、または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 AWS アカウント 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。
- Amazon EventBridge は、アプリケーションをさまざまなイベントソースのデータに簡単に接続できるようにするサーバーレスイベントバスサービスです。EventBridge は、お客様独自のアプリケーション、Software as a Service (SaaS) アプリケーション、AWS のサービスからのリアルタイムデータをストリーム配信し、そのデータを Lambda などのターゲットにルーティングします。これにより、サービスで発生したイベントをモニタリングし、イベント駆動型アーキテクチャを構築できます。詳細については、「[Amazon EventBridge ユーザーガイド](#)」および「[Amazon EventBridge イベントリファレンス](#)」を参照してください。
- AWS Billing and Cost Management は、請求書の支払いとコストの最適化に役立つ機能を提供するサービスです。アマゾン ウェブ サービスでは、使用量に基づいてアカウントに請求されるため、使用した分に対してのみ料金が発生します。AWS また、では、リソースに[コスト配分タグ](#)を適用して AWS コストを追跡および分類することもできます。たとえば、では AWS Clean Rooms、コラボレーションメンバーシップにタグを適用して、コラボレーションあたりのコストを追跡できます。詳細については、「[AWS Billing ユーザーガイド](#)」を参照してください。

分析ログイン AWS Clean Rooms

分析ログ記録は の機能です AWS Clean Rooms。 [コラボレーションを作成し](#)、分析ログ記録を有効にすると、メンバーは Amazon CloudWatch Logs のジョブからのクエリまたはログから関連するログを保存できます。

クエリログとジョブログを使用すると、メンバーはクエリが分析ルールに準拠しているかどうか、およびコラボレーション契約に準拠しているかどうかを判断できます。さらに、クエリログは監査にも役立ちます。

AWS Clean Rooms コンソールで分析ログ記録オプションが有効になっている場合、クエリログには以下が含まれます。

- `analysisRule` – 設定済みテーブルの分析ルール。
- `analysisTemplateArn` – 実行された分析テンプレート (分析ルールに応じて表示されます)。
- `collaborationId` – クエリが実行されたコラボレーションの一意の ID。
- `configuredTableID` – クエリで参照された設定済みテーブルの一意の ID。
- `directQueryAnalysisRulePolicy.custom.allowedAnalysis` – 設定済みテーブルで実行が許可された分析テンプレート (分析ルールに応じて表示されます)。
- `directQueryAnalysisRulePolicy.v1.custom.allowedAnalysisProviders` – クエリの作成を許可されたクエリプロバイダー (分析ルールに応じて表示されます)。
- `errorCode` – クエリが正しく実行されなかったときのエラーコード。
- `errorMessage` – クエリが正しく実行されなかったときのエラーメッセージ。
- `eventId` – 実行されたクエリの一意の ID。2023 年 8 月 31 日以降、この一意の ID は `protectedQueryID` と同じになります。
- `eventTimestamp` – クエリ実行時間。
- `parameters.parametervalue` – パラメータ値 (クエリテキストに応じて表示されます)。
- `queryText` – 実行されたクエリの SQL 定義。パラメータがある場合は、`:parametervalue` というラベルが付けられます。
- `queryValidationErrors` – クエリ検証時のクエリエラー。
- `schemaName` – クエリで参照された設定済みテーブルの関連付けの名前。
- `status` – クエリの実行ステータス。

クエリログとジョブログの受信

クエリログとジョブログをセットアップ AWS Clean Rooms するために、 の外部でアクションを実行する必要はありません。AWS Clean Rooms は、各コラボレーションメンバーが [メンバーシップを作成](#)した後に、コラボレーション用のロググループを作成します。

クエリを実行できるメンバー、クエリとジョブを実行できるメンバー、結果を受信できるメンバー、およびクエリで設定テーブルが参照されているメンバーには、クエリログまたはジョブログが送信されます。

クエリを行えるメンバーと結果を受け取れるメンバーは、クエリで参照される各設定済みテーブルのクエリログを受け取ります。双方が設定済みテーブルの所有者でない場合、設定済みテーブルの ID (configuredTableID) は表示されません。

クエリとジョブを実行できるメンバー、および結果を受け取ることができるメンバーは、ジョブで参照される設定済みテーブルごとにジョブログを受け取ります。双方が設定済みテーブルの所有者でない場合、設定済みテーブルの ID (configuredTableID) は表示されません。

1 人のメンバーが所有する複数の設定済みテーブルの関連付けがクエリで参照されている場合、そのメンバーは設定済みテーブルごとにクエリログを受け取ります。

メンバーにジョブで参照されている設定済みテーブルの関連付けが複数ある場合、設定済みテーブルごとにジョブログを受け取ります。

AWS Clean Roomsでサポートされていない、またはサポートされている SQL を含むクエリログが作成されます。詳細については、「[AWS Clean Rooms SQL リファレンス](#)」を参照してください。

ログは、クエリまたはジョブが、コラボレーションに関連付けられていない設定済みテーブルを参照するときにも作成されます。

ログには、誤った SQL に関する情報が含まれている可能性があります。

クエリログとジョブログはクエリのステータスを示しますが、クエリ出力が配信されたかどうかを報告しません。クエリまたはジョブが、クエリを実行できるメンバーによって送信されたことを確認します。クエリログは、クエリに でサポートされている SQL が含まれ AWS Clean Rooms 、コラボレーションに関連付けられた設定済みテーブルを参照していることも確認します。

Example

たとえば、クエリが分析ルールへの準拠 AWS Clean Rooms を検証した後、クエリ処理中にキャンセルされた場合、ログは生成されません。

ロググループを削除した場合は、同じロググループ名 (コラボレーションのコラボレーション ID) を使用してロググループを手動で再作成する必要があります。または、メンバーシップのログの有効/無効を切り替えることもできます。

分析ログ記録を有効にする方法の詳細については、「」を参照してください[コラボレーションの作成](#)。

Amazon CloudWatch Logs の詳細については、「[Amazon CloudWatch Logs ユーザーガイド](#)」を参照してください。

クエリログとジョブログに推奨されるアクション

メンバーが次のアクションを定期的に行うことを推奨します。

- クエリとジョブがコラボレーションで合意されたユースケースまたはクエリと一致することを確認するには、コラボレーションで実行されるクエリとジョブを確認します。

最近のクエリを表示する方法の詳細については、「[最近のクエリの表示](#)」を参照してください。

最近のジョブを表示する方法の詳細については、「」を参照してください[最近のジョブの表示](#)。

- 設定済みテーブルの列がコラボレーションに際して合意されたものと一致していることを確認するために、コラボレーションメンバーの分析ルールとクエリで使用する設定済みテーブルの列を確認してください。

設定した列を表示する方法の詳細については、「[テーブルと分析ルールの表示](#)」を参照してください。

での CloudWatch による詳細なモニタリングAWS Clean Rooms

詳細モニタリングはオプトイン機能ですAWS Clean Rooms。[コラボレーションを作成して](#)詳細モニタリングを有効にすると、クエリを実行できるメンバーと設定済みの支払者は、CloudWatch アカウントで詳細なオブザーバビリティメトリクスを受け取ることを選択できます。

詳細モニタリングを使用すると、メンバーはクエリパフォーマンスのモニタリング、リソース使用率の追跡、運用上のインサイトのためのダッシュボードの作成を行うことができます。これらのメトリクスは、キャパシティプランニング、コスト分析、パフォーマンス問題のトラブルシューティングに役立ちます。

Note

メトリクスの可用性は、設定されたテーブルデータソースによって異なります。

- Athena をデータソースとする設定済みテーブルを参照する分析の場合、BytesReadおよびQueryFileScannedメトリクスは常に 0 を返します。
- Snowflake をデータソースとして設定済みテーブルを参照する分析の場合、BytesRead、QueryFileScanned、および RecordsReadメトリクスは常に 0 を返します。
- 異なるデータソースを持つ複数の設定済みテーブル (Snowflake テーブルや S3-backedAWS Glue テーブルなど) を参照BytesReadし、S3-backedAWS Glue テーブルのメトリクスのみをQueryFileScanned返す分析の場合、は S3-backedAWS Glue テーブルと Athena テーブルのメトリクスRecordsReadを返します。

メトリクスの種類

AWS Clean Roomsは 2 つのカテゴリのメトリクスを CloudWatch に発行します。

集約された詳細モニタリングメトリクス

これらのメトリクスは、分析全体で集計レベルの可視性を提供します。

AnalysisRuntime

分析の合計時間をミリ秒単位で表示します。

CRPUConsumed

特定の分析に消費される CRPU (クリーンルーム処理ユニット)。CRPU の計算の詳細については、[AWS Clean Rooms「料金表」](#)を参照してください。

ConcurrentVCPUs

特定の時点で使用されている同時 vCPU (サービスクォータ関連)。

AnalysesCount

指定されたディメンションに対して一定期間実行される分析の数。

ConcurrentQueries

特定の時点での同時クエリ数 (サービスクォータ関連)。

クエリレベルの詳細モニタリングメトリクス

これらのメトリクスは、個々のクエリ実行に関する詳細なインサイトを提供します。

BytesRead

分析のためにソーステーブルから読み取られた合計データ (バイト単位)。

BytesWritten

分析用に書き込まれた合計データ (バイト単位)。

RecordsRead

分析のためにソーステーブルから読み取られたレコードの合計数。

RecordsWritten

分析結果で返されたレコードの合計数。

QueryFileScanned

分析用にスキャンされたデータファイルの合計数。

Memory utilization

分析に使用されるメモリリソース。

vCPU utilization

分析に使用される vCPU リソース。

Disk utilization

分析に使用されるストレージリソース。

メトリクスのディメンション

メトリクスは、次のディメンションによって整理されます。

- 分析ステータス: 完了
- 分析タイプ: SparkSQL
- メンバーシップ: メンバーシップ ID
- クエリ: 個々のクエリ ID (クエリレベルのメトリクス用)
- アカウント/リージョン: AWS アカウントと リージョン

すべてのメトリクスは CloudWatch AWS/CleanRooms の名前空間に発行されます。

メトリクスにアクセスできるユーザー

詳細なモニタリングメトリクスは、以下に使用できます。

- クエリランナー: コラボレーションでクエリを実行できるメンバー
- 設定済み支払者: クエリ実行コストを支払うように指定されたメンバー

データのみを提供するメンバーは、詳細なモニタリングメトリクスにアクセスできません。

料金

AWS Clean Rooms は CloudWatch へのメトリクスの発行には課金しません。ただし、メトリクスストレージ、ダッシュボードの使用、アラームには標準の CloudWatch 料金が適用されます。詳細については、[Amazon CloudWatch 料金表](#)をご覧ください。

を使用した AWS Clean Rooms API コールのログ記録 AWS CloudTrail

AWS Clean Rooms は、ユーザー AWS CloudTrail、ロール、または によって実行されたアクションを記録するサービスである と統合 AWS のサービス されています AWS Clean Rooms。CloudTrail は、 のすべての API コールをイベント AWS Clean Rooms 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 Clean Rooms コンソールからの呼び出しと AWS Clean Rooms 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、イベントを含む 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 AWS Clean Rooms。追跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。CloudTrail が収集した情報を使用して、AWS Clean Rooms に対して行われた要求、要求が行われた IP アドレス、要求を行った人、要求が行われた日時、および追加の詳細を判別できます。

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 Clean Rooms CloudTrail の情報

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。でアクティビティが発生すると AWS Clean Rooms、そのアクティビティはイベント履歴の他の AWS のサービス イベント

トとともに CloudTrail イベントに記録されます。最近のイベントは、AWS アカウントで表示、検索、ダウンロードできます。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。

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 Clean Rooms、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より詳細に分析し、それを基にアクションを取るために他の AWS の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](#)」
- [CloudTrail ログファイルの複数のリージョンからの受け取り](#)
- [複数のアカウントから CloudTrail ログファイルを受け取る](#)

すべての AWS Clean Rooms アクションは CloudTrail によってログに記録され、[AWS Clean Rooms API リファレンス](#)に記載されています。

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、ルートユーザーまたは IAM ユーザーのどちらの認証情報を使用して送信されたかどうか。
- 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 のサービスによって送信されたかどうか。

詳細については、「[CloudTrail userIdentity エレメント](#)」を参照してください。

AWS Clean Rooms 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 呼び出し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。

Example AWS Clean Rooms CloudTrail イベント

次の例は、以下の CloudTrail イベントを示しています。

トピック

- [StartProtectedQuery \(成功\)](#)
- [StartProtectedQuery \(失敗\)](#)

StartProtectedQuery (成功)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:sts::123456789012:assumed-role/query-runner/jdoe",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/query-runner",
        "accountId": "123456789012",
        "userName": "query-runner"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-04-07T19:34:32Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-04-07T19:53:32Z",
  "eventSource": "cleanrooms.amazonaws.com",
  "eventName": "StartProtectedQuery",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.1",
```



```

    "userAgent": "aws-internal/3",
    "requestParameters": {
      "resultConfiguration": {
        "outputConfiguration": {
          "s3": {
            "resultFormat": "CSV",
            "bucket": "cleanrooms-queryresults-jdoe-test",
            "keyPrefix": "test"
          }
        }
      },
      "sqlParameters": "****",
      "membershipIdentifier": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "type": "SQL"
    },
    "responseElements": {
      "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-ErrorMessage,Date",
      "protectedQuery": {
        "createTime": 1680897212.279,
        "id": "f5988bf1-771a-4141-82a8-26fcc4e41c9f",
        "membershipArn": "arn:aws:cleanrooms:us-east-2:123456789012:membership/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "membershipId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resultConfiguration": {
          "outputConfiguration": {
            "s3": {
              "bucket": "cleanrooms-queryresults-jdoe-test",
              "keyPrefix": "test",
              "resultFormat": "CSV"
            }
          }
        },
        "sqlParameters": "****",
        "status": "SUBMITTED"
      }
    },
    "requestID": "7464211b-2277-4b55-9723-fb4f259aefd2",
    "eventID": "f7610f5e-74b9-420f-ae43-206571ebcbf7",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"

```

```
}
```

StartProtectedQuery (失敗)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:sts::123456789012:assumed-role/query-runner/jdoe",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/query-runner",
        "accountId": "123456789012",
        "userName": "query-runner"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-04-07T19:34:32Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-04-07T19:47:27Z",
  "eventSource": "cleanrooms.amazonaws.com",
  "eventName": "StartProtectedQuery",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.1",
  "userAgent": "aws-internal/3",
  "errorCode": "ValidationException",
  "requestParameters": {
    "resultConfiguration": {
      "outputConfiguration": {
        "s3": {
          "resultFormat": "CSV",
          "bucket": "cleanrooms-queryresults-jdoe-test",
          "keyPrefix": "test"
        }
      }
    }
  }
}
```

```

    },
    "sqlParameters": "****",
    "membershipIdentifier": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "type": "SQL"
  },
  "responseElements": {
    "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-ErrorMessage,Date",
    "message": "Column(s) [identifier] is not allowed in select"
  },
  "requestID": "e29f9f74-8299-4a83-9d18-5ddce7302f07",
  "eventID": "c8ee3498-8e4e-44b5-87e4-ab9477e56eb5",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}

```

Amazon EventBridge を使用してイベント駆動型アプリケーション AWS Clean Rooms に統合する

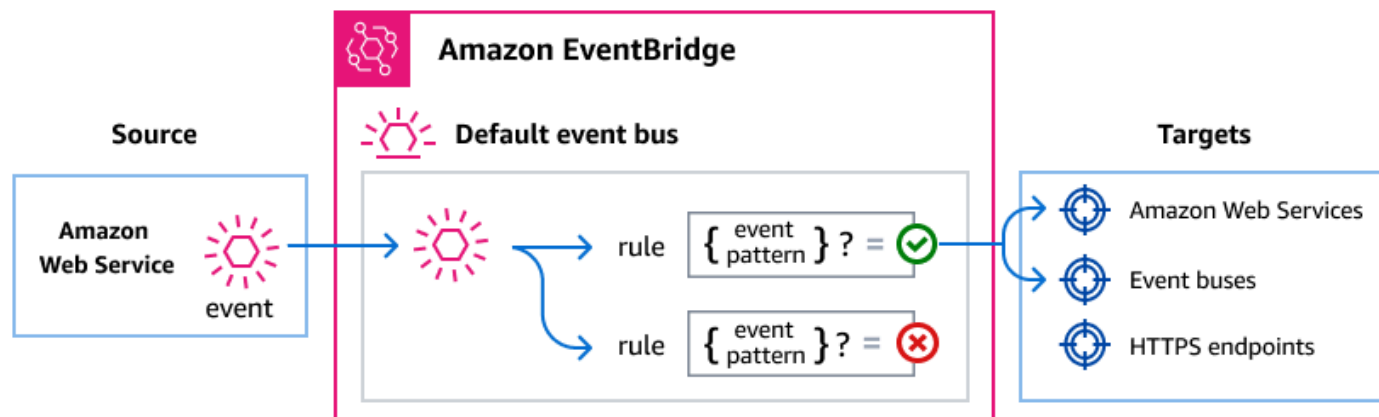
で発生するイベントを使用してアプリケーションコンポーネント間で通信し、ダウンストリームプロセスを開始するイベント駆動型アプリケーション (EDAs) AWS Clean Rooms AWS Clean Rooms に組み込むことができます。これを行うには、Amazon EventBridge を使用して、 から他のソフトウェアコンポーネントにイベント AWS Clean Rooms をルーティングします。Amazon EventBridge は、イベントを使用してアプリケーションコンポーネントを接続するサーバーレスサービスです。これにより、コードやオペレーションを追加することなく、 のような AWS AWS Clean Rooms サービスをイベント駆動型アーキテクチャに簡単に統合できます。

イベント駆動型アーキテクチャとは、イベントの発信と応答によって連携する、疎結合のソフトウェアシステムを構築するスタイルです。このモデルでは、イベントはリソースまたは環境の変更を表します。

EventBridge の仕組みは次のとおりです AWS Clean Rooms。

多くの AWS サービスと同様に、 はイベント AWS Clean Rooms を生成し、EventBridge のデフォルトイベントバスに送信します。イベントバスはルーターであり、イベントを受信して指定された送信先やターゲットに配信します。ターゲットには、他の AWS サービス、カスタムアプリケーション、SaaS パートナーアプリケーションを含めることができます。

EventBridge は、イベントバスで作成されたルールに従ってイベントを配信します。ルールごとに、フィルターまたはイベントパターンを指定して、必要なイベントのみを選択します。イベントがイベントバスに送信されるたびに、EventBridge はイベントを各ルールと比較します。イベントがルールに一致すると、EventBridge は指定されたターゲットにイベントを配信します。



例えば、アカウントで新しい AWS Clean Rooms コラボレーションが作成されるたびに知っておく必要があるとします。デフォルトのイベントバスにルールを作成できます。ルールでは、**Collaboration Created** という名前の AWS クリーンルームからのイベントを指定するイベントパターンを作成します。EventBridge は、これらのプロパティに一致するイベントを受信するたびに、指定されたワークフローにイベントをルーティングします。

AWS Clean Rooms イベント

AWS サービスは、EventBridge のデフォルトのイベントバスに直接イベントを送信できます。さらに、AWS CloudTrail は多数の AWS サービスから発生したイベントも EventBridge に送信します。これらのイベントには、API コール、コンソールのサインインとアクション、サービスイベント、CloudTrail Insights などがあります。詳細については、「Amazon EventBridge ユーザーガイド」の「[AWS service events delivered via AWS CloudTrail](#)」を参照してください。

EventBridge に送信される AWS Clean Rooms イベントの完全なリストについては、[EventBridge イベントリファレンス](#)の AWS Clean Rooms トピックを参照してください。

イベントの詳細のタイプ	説明
作成された分析テンプレート	分析テンプレートの作成時に、分析テンプレートの所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。

イベントの詳細のタイプ	説明
分析テンプレートが更新されました	分析テンプレートの所有者と、更新を確認できるコラボレーションのすべてのアクティブなメンバーは、分析テンプレートが更新されると通知されます。
削除された分析テンプレート	分析テンプレートの所有者とコラボレーションのすべてのアクティブなメンバーは、分析テンプレートが削除されると通知されます。
作成されたコラボレーション	コラボレーションの作成時にコラボレーション所有者に通知されます。
コラボレーションの更新	コラボレーション所有者と、コラボレーションの更新を可視化できるコラボレーションのすべてのアクティブなメンバーは、コラボレーションが更新されると通知されます。
作成されたコラボレーション変更リクエスト	コラボレーション変更リクエストが作成されると、コラボレーション所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。
コラボレーション変更リクエストが承認されました	コラボレーション変更リクエストが承認されると、コラボレーション所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。
コラボレーション変更リクエストがキャンセルされました	コラボレーション変更リクエストがキャンセルされると、コラボレーション所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。
コラボレーション変更リクエストのコミット	コラボレーション変更リクエストがコミットされると、コラボレーション所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。
設定済みテーブルの関連付けが作成されました	設定済みテーブルの関連付けの所有者とコラボレーションのすべてのアクティブなメンバーは、設定済みテーブルの関連付けが作成されると通知されます。

イベントの詳細のタイプ	説明
設定済みテーブルの関連付けが更新されました	設定済みテーブルの関連付けの所有者と、更新を確認できるコラボレーションのすべてのアクティブなメンバーは、設定済みテーブルの関連付けが更新されると通知されます。
設定済みテーブルの関連付けが削除されました	設定済みテーブルの関連付けの所有者とコラボレーションのすべてのアクティブなメンバーは、設定済みテーブルの関連付けが削除されると通知されます。
設定済みテーブルの関連付け分析ルールが作成されました	設定済みテーブル関連付け分析ルールの所有者とコラボレーションのすべてのアクティブなメンバーは、設定済みテーブル関連付け分析ルールが作成されると通知されます。
設定済みテーブルの関連付け分析ルールが更新されました	設定済みテーブルの関連付け分析ルールの所有者と、更新を確認できるコラボレーションのすべてのアクティブなメンバーは、設定済みテーブルの関連付け分析ルールが更新されると通知されます。
設定済みテーブルの関連付け分析ルールが削除されました	設定済みテーブル関連付け分析ルールの所有者とコラボレーションのすべてのアクティブなメンバーは、設定済みテーブル関連付け分析ルールが削除されると通知されます。
ID マッピングテーブルが作成されました	ID マッピングテーブルの所有者とコラボレーションのすべてのアクティブなメンバーは、ID マッピングテーブルが作成されると通知されます。
ID マッピングテーブルが更新されました	ID マッピングテーブルの所有者と、更新を確認できるコラボレーションのすべてのアクティブなメンバーは、ID マッピングテーブルが更新されると通知されます。
ID マッピングテーブルが削除されました	ID マッピングテーブルの所有者とコラボレーションのすべてのアクティブなメンバーは、ID マッピングテーブルが削除されると通知されます。
ID 名前空間の関連付けが作成されました	ID 名前空間関連付けの所有者とコラボレーションのすべてのアクティブなメンバーは、ID 名前空間関連付けが作成されると通知されます。

イベントの詳細のタイプ	説明
ID 名前空間の関連付けが更新されました	ID 名前空間関連付けの所有者と、更新を確認できるコラボレーションのすべてのアクティブなメンバーは、ID 名前空間関連付けが更新されると通知されます。
ID 名前空間の関連付けが削除されました	ID 名前空間関連付けの所有者とコラボレーションのすべてのアクティブなメンバーは、ID 名前空間関連付けが削除されると通知されます。
コラボレーションに招待されました	招待されたメンバーは、コラボレーションに招待されると通知されます。
作成されたメンバーシップ	メンバーシップの作成時に、メンバーシップ所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。
メンバーシップの更新	メンバーシップが更新されると、メンバーシップ所有者に通知されます。ただし、メンバーシップがコラボレーションから削除された場合は、コラボレーションのすべてのアクティブなメンバーに通知されます。
削除されたメンバーシップ	メンバーシップが削除されると、メンバーシップ所有者とコラボレーションのすべてのアクティブなメンバーに通知されます。
保護されたジョブが送信されました	保護対象ジョブの送信時に、保護対象ジョブのジョブランナー、ジョブ支払者、結果レシーバーに通知されます。
保護ジョブが開始されました	保護対象ジョブのジョブランナー、ジョブ支払者、結果レシーバーは、保護対象ジョブの開始時に通知されます。
保護されたジョブのキャンセル	保護されたジョブのジョブランナー、ジョブ支払者、結果レシーバーは、保護されたジョブがキャンセルされると通知されます。
保護されたジョブがキャンセルされました	保護対象ジョブのジョブランナー、ジョブ支払者、結果レシーバーは、保護対象ジョブがキャンセルされると通知されます。

イベントの詳細のタイプ	説明
保護されたジョブが成功しました	保護対象ジョブのジョブランナー、ジョブ支払者、結果レシーバーは、保護対象ジョブが成功すると通知されます。
保護ジョブが失敗しました	保護対象ジョブのジョブランナー、ジョブ支払者、結果レシーバーは、保護対象ジョブが失敗すると通知されます。
保護されたクエリが送信されました	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリが送信されると通知されます。
保護されたクエリが開始されました	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリが開始されると通知されます。
保護されたクエリのキャンセル	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリがキャンセルされると通知されます。
保護されたクエリがキャンセルされました	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリがキャンセルされると通知されます。
保護されたクエリが成功しました	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリが成功すると通知されます。
保護されたクエリが失敗しました	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリが失敗すると通知されます。
保護されたクエリがタイムアウトしました	保護されたクエリのクエリランナー、クエリ支払者、結果レシーバー (複数可) は、保護されたクエリがタイムアウトすると通知されます。

EventBridge を使用した AWS Clean Rooms イベントのルーティング

EventBridge で AWS Clean Rooms イベントをターゲットにルーティングするには、ルールを作成する必要があります。各ルールにはイベントパターンが含まれており、EventBridge はイベントバスで受信した各イベントと照合します。イベントデータが指定されたイベントパターンと一致する場合、EventBridge はそのイベントをルールのターゲット (複数可) にルーティングします。

イベントバスルールの詳細な作成方法については、「EventBridge ユーザーガイド」の「[Creating rules that react to events](#)」を参照してください。

イベントに一致する AWS Clean Rooms イベントパターンの作成

各イベントパターンは JSON 形式のオブジェクトで、以下が含まれています。

- (オプション): イベントを送信するサービスを識別する `source` 属性。AWS Clean Rooms イベントの場合、ソースは `aws.cleanrooms` です。
- (オプション): 照合するイベント名の配列を含む `detail-type` 属性。
- (オプション): 照合対象となるその他のイベントデータを含む `detail` 属性。

たとえば、次のイベントパターンは、コラボレーションが削除されたすべての Id 名前空間関連付けの更新済みイベントと一致します AWS Clean Rooms。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Namespace Association Updated"],
  "detail": {
    "status": ["COLLABORATION_DELETED"]
  }
}
```

イベントパターンの記述の詳細については、「EventBridge ユーザーガイド」の「[Event patterns](#)」を参照してください。

AWS Clean Rooms イベント詳細リファレンス

AWS サービスからのすべてのイベントには、イベントのソースである AWS サービス、イベントが生成された時刻、イベントが発生したアカウントとリージョンなど、イベントに関するメタデータを含む共通のフィールドセットがあります。これらの一般的なフィールドの定義については、「Amazon EventBridge イベントリファレンス」の「[イベントの構造](#)」を参照してください。

さらに、各イベントには、その特定のイベントに固有のデータを含む detail フィールドがあります。以下のリファレンスでは、さまざまな AWS Clean Rooms イベントの詳細フィールドを定義しています。

EventBridge を使用して AWS Clean Rooms イベントを選択および管理する場合、次の点に注意してください。

- からのすべてのイベントの source フィールド AWS Clean Rooms は に設定されます `aws.cleanrooms`。
- detail-type フィールドはイベントタイプを指定します。

例えば、Collaboration Created。

- detail フィールドには、その特定のイベントに固有のデータが含まれます。

ルールがイベントを照合 AWS Clean Rooms できるようにするイベントパターンの構築については、「Amazon EventBridge ユーザーガイド」の [「イベントパターン」](#) を参照してください。

分析テンプレート作成イベント

以下は、Analysis Template Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Analysis Template Created"]
}
```

分析テンプレートの更新イベント

以下は、Analysis Template Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Analysis Template Updated"]
}
```

分析テンプレートの削除イベント

以下は、Analysis Template Deleted イベントの詳細フィールドです。

```
{
```

```
"source": ["aws.cleanrooms"],
"detail-type": ["Analysis Template Deleted"]
}
```

コラボレーション作成イベント

以下は、Collaboration Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Collaboration Created"]
}
```

コラボレーションの更新イベント

以下は、Collaboration Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Collaboration Updated"]
}
```

コラボレーション変更リクエスト作成イベント

以下は、Collaboration Change Request Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Collaboration Change Request Created"]
}
```

コラボレーション変更リクエスト承認済みイベント

以下は、Collaboration Change Request Approved イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Collaboration Change Request Approved"]
}
```

コラボレーション変更リクエストのキャンセルイベント

以下は、Collaboration Change Request Cancelled イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Collaboration Change Request Cancelled"]
}
```

コラボレーション変更リクエストのコミットイベント

以下は、Collaboration Change Request Committed イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Collaboration Change Request Committed"]
}
```

設定済みテーブルの関連付け作成イベント

以下は、Configured Table Association Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Configured Table Association Created"]
}
```

設定済みテーブルの関連付けの更新イベント

以下は、Configured Table Association Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Configured Table Association Updated"]
}
```

設定済みテーブルの関連付け削除イベント

以下は、Configured Table Association Deleted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Configured Table Association Deleted"]
}
```

設定済みテーブル関連付け分析ルール作成イベント

以下は、Configured Table Association Analysis Rule Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Configured Table Association Analysis Rule Created"]
}
```

設定済みテーブル関連付け分析ルールの更新イベント

以下は、Configured Table Association Analysis Rule Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Configured Table Association Analysis Rule Updated"]
}
```

設定済みテーブル関連付け分析ルールの削除イベント

以下は、Configured Table Association Analysis Rule Deleted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Configured Table Association Analysis Rule Deleted"]
}
```

ID マッピングテーブル作成イベント

以下は、Id Mapping Table Created イベントの詳細フィールドです。

```
{
```

```
"source": ["aws.cleanrooms"],
"detail-type": ["Id Mapping Table Created"]
}
```

ID マッピングテーブルの更新イベント

以下は、Id Mapping Table Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Mapping Table Updated"]
}
```

ID マッピングテーブルの削除イベント

以下は、Id Mapping Table Deleted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Mapping Table Deleted"]
}
```

ID 名前空間の関連付け作成イベント

以下は、Id Namespace Association Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Namespace Association Created"]
}
```

ID 名前空間の関連付けの更新イベント

以下は、Id Namespace Association Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Namespace Association Updated"]
}
```

 Note

コラボレーションが削除されると、イベントは `detail.status` フィールドの ID 名前空間関連付けの更新イベントにキャプチャされます。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Namespace Association Updated"],
  "detail": {
    "status": ["COLLABORATION_DELETED"]
  }
}
```

ID 名前空間の関連付けが削除されたイベント

以下は、Id Namespace Association Deleted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Id Namespace Association Deleted"]
}
```

コラボレーションイベントに招待されました

以下は、Invited To Collaboration イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Invited To Collaboration"]
}
```

メンバーシップ作成イベント

以下は、Membership Created イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Membership Created"]
}
```

メンバーシップ更新イベント

以下は、Membership Updated イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Membership Updated"]
}
```

メンバーシップ削除イベント

以下は、Membership Deleted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Membership Deleted"]
}
```

保護されたジョブの送信イベント

以下は、Protected Job Submitted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Job Submitted"]
}
```

保護ジョブ開始イベント

以下は、Protected Job Started イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Job Started"]
}
```

保護されたジョブキャンセルイベント

以下は、Protected Job Cancelling イベントの詳細フィールドです。


```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Job Cancelling"]
}
```

保護されたジョブのキャンセルイベント

以下は、Protected Job Cancelled イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Job Cancelled"]
}
```

保護されたジョブの成功イベント

以下は、Protected Job Succeeded イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Job Succeeded"]
}
```

保護ジョブ失敗イベント

以下は、Protected Job Failed イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Job Failed"]
}
```

保護されたクエリ送信イベント

以下は、Protected Query Submitted イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Submitted"]
}
```

保護されたクエリ開始イベント

以下は、Protected Query Started イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Started"]
}
```

保護されたクエリキャンセルイベント

以下は、Protected Query Cancelling イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Cancelling"]
}
```

保護されたクエリのキャンセルイベント

以下は、Protected Query Cancelled イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Cancelled"]
}
```

保護されたクエリ成功イベント

以下は、Protected Query Succeeded イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Succeeded"]
}
```

保護されたクエリ失敗イベント

以下は、Protected Query Failed イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Failed"]
}
```

保護されたクエリのタイムアウトイベント

以下は、Protected Query Timed Out イベントの詳細フィールドです。

```
{
  "source": ["aws.cleanrooms"],
  "detail-type": ["Protected Query Timed Out"]
}
```

コスト配分タグ

AWS Clean Rooms では、コスト配分タグを使用して AWS コストを追跡できます。AWS Billing and Cost Management ダッシュボードでこれらのタグをアクティブ化します。はタグ AWS を使用してコストを分類し、毎月のコスト配分レポートを配信します。ユーザー定義のコスト配分タグを AWS Clean Rooms リソースに適用して、コラボレーション全体のコストを追跡して割り当てることができます。

コスト配分のタグ付け可能なリソース

次の表は、料金が発生し、コストを追跡および整理するためにユーザー定義のコスト配分タグでタグ付けできる AWS Clean Rooms リソースの一覧です。

請求対象リソース	タグ付きリソース
SQL クエリ	メンバーシップ
PySpark ジョブ	メンバーシップ
ML トレーニングジョブ	メンバーシップ
ML 推論ジョブ	メンバーシップ
合成データ生成	メンバーシップ

請求対象リソース	タグ付きリソース
類似モデルトレーニング	類似モデル
類似セグメントのエクスポート	類似セグメント

を使用した AWS Clean Rooms リソースの作成 AWS CloudFormation

AWS Clean Rooms は AWS CloudFormation、AWS リソースのモデル化とセットアップに役立つサービスであると統合されています。この統合により、リソースとインフラストラクチャの作成、管理に費やす時間を短縮できます。必要なすべてのリソースを AWS 記述するテンプレートを作成し、それらのリソースを CloudFormation プロビジョニングして設定します。リソースの例には、コラボレーション、設定済みテーブル、設定済みテーブルの関連付け、メンバーシップなどがあります。

を使用すると CloudFormation、テンプレートを再利用して AWS Clean Rooms リソースを一貫して繰り返しセットアップできます。リソースを一度記述し、同じリソースを複数の AWS アカウントとで何度もプロビジョニングします AWS リージョン。

AWS Clean Rooms および CloudFormation テンプレート

および関連サービスのリソースをプロビジョニング AWS Clean Rooms および設定するには、[CloudFormation テンプレート](#)を理解する必要があります。テンプレートは、JSON や YAML でフォーマットされたテキストファイルです。これらのテンプレートは、CloudFormation スタックでプロビジョニングするリソースを記述します。JSON または YAML に慣れていない場合は、CloudFormation デザイナーを使用して CloudFormation テンプレートの使用を開始できます。詳細については、「AWS CloudFormation ユーザーガイド」の「[CloudFormation Designer とは](#)」を参照してください。

AWS Clean Rooms は、でのコラボレーション、設定済みテーブル、設定済みテーブルの関連付け、メンバーシップの作成をサポートしています CloudFormation。コラボレーション用の JSON テンプレートと YAML テンプレートの例、設定済みテーブル、設定済みテーブルの関連付け、メンバーシップなどの詳細については、AWS CloudFormation 「ユーザーガイド」の「」および[AWS Clean Rooms 「AWS Clean Rooms ML](#) リソースタイプのリファレンス」を参照してください。

の詳細 CloudFormation

詳細については CloudFormation、次のリソースを参照してください。

- [AWS CloudFormation](#)
- [AWS CloudFormation ユーザーガイド](#)

- [CloudFormation API リファレンス](#)
- [AWS CloudFormation コマンドラインインターフェイスユーザーガイド](#)

のクォータ AWS Clean Rooms

AWS アカウント には、各 の制限と呼ばれるデフォルトのクォータがあります AWS のサービス。特に明記されていない限り、各クォータは に固有です AWS リージョン。一部のクォータの増加を要求できますが、他のクォータは増加できません。

のクォータを表示するには AWS Clean Rooms、[Service Quotas コンソール](#)を開きます。ナビゲーションペインで、[AWS のサービス] を選択し、[AWS Clean Rooms] 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 でまだクォータが利用できない場合は、[サービス制限の緩和フォーム](#)を使用してください。

トピック

- [AWS Clean Rooms クォータ](#)
- [AWS Clean Rooms ML クォータ](#)

AWS Clean Rooms クォータ

AWS アカウント には、次のクォータが関連しています AWS Clean Rooms。

名前	デフォルト	引き上げ可能	説明
分析ルールサイズ	サポートされている各リージョン: 100 KB	いいえ	分析ルールの JSON の最大サイズ
メンバーシップあたりの分析テンプレート	サポートされている各リージョン: 25	可能	メンバーシップあたりの分析テンプレートの最大数

名前	デフォルト	引き上げ可能	説明
アカウントあたりの作成済みコラボレーション	サポートされている各リージョン: 10	あり	各アカウントで作成されるコラボレーションの最大数
設定済みテーブル許可リストあたりの列	サポートされている各リージョン: 100	可能	設定された各テーブルでホワイトリストに登録できる列の最大数
アカウントあたりの PySpark ジョブ vCPU の同時使用量	us-east-1: 1,024 サポートされている他の各リージョン: 512	あり	アカウントあたりのすべての同時実行 PySpark ジョブの最大合計 vCPU 使用量
アカウントあたりの同時 PySpark ジョブ	us-east-1: 5 他のサポートされている各リージョン: 2	あり	アカウントあたりの同時実行 PySpark ジョブの最大数
アカウントあたりの同時 SQL クエリ	us-east-1: 5 他のサポートされている各リージョン: 2	あり	アカウントあたりの同時実行 SQL クエリの最大数
アカウントあたりの同時 SQL クエリ vCPU 使用量	サポートされている各リージョン: 512	あり	アカウントあたりのすべての同時実行 SQL クエリの最大合計 vCPU 使用量

名前	デフォルト	引き上げ可能	説明
メンバーシップあたりの同時進行ジョブ	サポートされている各リージョン: 1	[いいえ]	メンバーシップあたりの同時進行ジョブの最大数
メンバーシップあたりの同時実行クエリ	サポートされている各リージョン: 5	いいえ	メンバーシップあたりの継続的な同時クエリの最大数
メンバーシップあたりの設定済みオーディエンスモデルの関連付け	サポートされている各リージョン: 5	あり	メンバーシップあたりの設定済みオーディエンスモデルの関連付けの最大数
アカウントあたりの設定済みテーブル	サポートされている各リージョン: 250	あり	各アカウントで作成される設定済みテーブルの最大数
保護されたクエリあたりの設定済みテーブル	サポートされている各リージョン: 15	可能	保護されたクエリで設定されたテーブルの最大数
メンバーシップあたりの ID マッピングテーブル	サポートされている各リージョン: 5	あり	メンバーシップあたりの ID マッピングテーブルの最大数
メンバーシップあたりの ID 名前空間の関連付け	サポートされている各リージョン: 10	あり	メンバーシップあたりの ID 名前空間の関連付けの最大数

名前	デフォルト	引き上げ可能	説明
コラボレーションあたりの招待済みメンバー	サポートされている各リージョン: 5	あり	各コラボレーションで招待されるメンバーの最大数
アカウントあたりのメンバーシップ	サポートされている各リージョン: 100	可能	各アカウントのメンバーシップの最大数
コラボレーションあたりのアクセス予算のプライバシー予算テンプレート	サポートされている各リージョン: 25	可能	コラボレーションあたりのアクセス予算のプライバシー予算テンプレートの最大数
クエリテキストの長さ (差分プライバシーを使用)	サポートされている各リージョン: 8 キロバイト	いいえ	差分プライバシーを使用する SQL クエリステートメントの最大テキスト長
Clean Rooms SQL 分析エンジンのクエリテキストの長さ	サポートされている各リージョン: 90 KB	いいえ	Clean Rooms SQL 分析エンジンの SQL クエリステートメントの最大テキスト長
Spark 分析エンジンのクエリテキストの長さ	サポートされている各リージョン: 500 キロバイト	いいえ	Spark 分析エンジンの SQL クエリステートメントの最大テキスト長
BatchGetCollaborationAnalysisTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの BatchGetCollaborationAnalysisTemplate API コールの最大数

名前	デフォルト	引き上げ可能	説明
BatchGetSchema リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの BatchGetSchema API 呼び出しの最大数
BatchGetSchemaAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの BatchGetSchemaAnalysisRule API コールの最大数
CreateAnalysisTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateAnalysisTemplate API コールの最大数
CreateCollaboration リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateCollaboration API 呼び出しの最大数
CreateCollaborationChangeRequest リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateCollaborationChangeRequest API コールの最大数
CreateConfiguredAudienceModelAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateConfiguredAudienceModelAssociation API コールの最大数
CreateConfiguredTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateConfiguredTable API 呼び出しの最大数

名前	デフォルト	引き上げ可能	説明
CreateConfiguredTableAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateConfiguredTableAnalysisRule API 呼び出しの最大数
CreateConfiguredTableAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateConfiguredTableAssociation API 呼び出しの最大数
CreateConfiguredTableAssociationAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateConfiguredTableAssociationAnalysisRule API コールの最大数
CreateIdMappingTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateIdMappingTable API コールの最大数
CreateIdNamespaceAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateIdNamespaceAssociation API コールの最大数
CreateMembership リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreateMembership API 呼び出しの最大数
CreatePrivacyBudgetTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの CreatePrivacyBudgetTemplate API コールの最大数

名前	デフォルト	引き上げ可能	説明
DeleteAnalysisTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteAnalysisTemplate API コールの最大数
DeleteCollaboration リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteCollaboration API 呼び出しの最大数
DeleteConfiguredAudienceModelAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteConfiguredAudienceModelAssociation API コールの最大数
DeleteConfiguredTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteConfiguredTable API 呼び出しの最大数
DeleteConfiguredTableAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteConfiguredTableAnalysisRule API 呼び出しの最大数
DeleteConfiguredTableAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteConfiguredTableAssociation API 呼び出しの最大数
DeleteConfiguredTableAssociationAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteConfiguredTableAssociationAnalysisRule API コールの最大数

名前	デフォルト	引き上げ可能	説明
DeleteldMappingTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteldMappingTable API コールの最大数
DeleteldNamespaceAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteldNamespaceAssociation API コールの最大数
DeleteMember リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteMember API 呼び出しの最大数
DeleteMembership リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeleteMembership API 呼び出しの最大数
DeletePrivacyBudgetTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの DeletePrivacyBudgetTemplate API コールの最大数
GetAnalysisTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetAnalysisTemplate API コールの最大数
GetCollaboration リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetCollaboration API 呼び出しの最大数
GetCollaborationAnalysisTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetCollaborationAnalysisTemplate API コールの最大数

名前	デフォルト	引き上げ可能	説明
GetCollaborationChangeRequest リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetCollaborationChangeRequest API コールの最大数
GetCollaborationConfiguredAudienceModelAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetCollaborationConfiguredAudienceModelAssociation API コールの最大数
GetCollaborationIdNamespaceAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetCollaborationIdNamespaceAssociation API コールの最大数
GetCollaborationPrivacyBudgetTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetCollaborationPrivacyBudgetTemplate API コールの最大数
GetConfiguredAudienceModelAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetConfiguredAudienceModelAssociation API コールの最大数
GetConfiguredTable リクエストのレート	サポートされている各リージョン: 20/秒	あり	1 秒あたりの GetConfiguredTable API 呼び出しの最大数
GetConfiguredTableAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetConfiguredTableAnalysisRule API 呼び出しの最大数

名前	デフォルト	引き上げ可能	説明
GetConfiguredTableAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetConfiguredTableAssociation API 呼び出しの最大数
GetConfiguredTableAssociationAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetConfiguredTableAssociationAnalysisRule API コールの最大数
GetIdMappingTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetIdMappingTable API コールの最大数
GetIdNamespaceAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetIdNamespaceAssociation API コールの最大数
GetMembership リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetMembership API 呼び出しの最大数
GetPrivacyBudgetTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetPrivacyBudgetTemplate API コールの最大数
GetProtectedJob リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetProtectedJob API コールの最大数

名前	デフォルト	引き上げ可能	説明
GetProtectedQuery リクエストのレート	サポートされている各リージョン: 20/秒	あり	1 秒あたりの GetProtectedQuery API 呼び出しの最大数
GetSchema リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetSchema API 呼び出しの最大数
GetSchemaAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの GetSchemaAnalysisRule API 呼び出しの最大数
ListAnalysisTemplates リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListAnalysisTemplates API コールの最大数
ListCollaborationAnalysisTemplates リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborationAnalysisTemplates API コールの最大数
ListCollaborationChangeRequests リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborationChangeRequests API コールの最大数
ListCollaborationConfiguredAudienceModelAssociations リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborationConfiguredAudienceModelAssociations API コールの最大数

名前	デフォルト	引き上げ可能	説明
ListCollaborationIdNamespaceAssociations リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborationIdNamespaceAssociations API コールの最大数
ListCollaborationPrivacyBudgetTemplates リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborationPrivacyBudgetTemplates API コールの最大数
ListCollaborationPrivacyBudgets リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborationPrivacyBudgets API コールの最大数
ListCollaborations リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListCollaborations API 呼び出しの最大数
ListConfiguredAudienceModelAssociations リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListConfiguredAudienceModelAssociations API コールの最大数
ListConfiguredTableAssociations リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListConfiguredTableAssociations API 呼び出しの最大数
ListConfiguredTables リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListConfiguredTables API 呼び出しの最大数

名前	デフォルト	引き上げ可能	説明
ListIdMappingTables リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListIdMappingTables API コールの最大数
ListIdNamespaceAssociations リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListIdNamespaceAssociations API コールの最大数
ListMembers リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListMembers API 呼び出しの最大数
ListMemberships リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListMemberships API 呼び出しの最大数
ListPrivacyBudgetTemplates リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListPrivacyBudgetTemplates API コールの最大数
ListPrivacyBudgets リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListPrivacyBudgets API コールの最大数
ListProtectedJobs リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListProtectedJobs API コールの最大数
ListProtectedQueries リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListProtectedQueries API 呼び出しの最大数

名前	デフォルト	引き上げ可能	説明
ListSchemas リクエストのレート	サポートされている各リージョン: 5	あり	1 秒あたりの ListSchemas API 呼び出しの最大数
ListTagsForResource リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの ListTagsForResource API コールの最大数
PopulateIdMappingTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの PopulateIdMappingTable API コールの最大数
PreviewPrivacyImpact リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの PreviewPrivacyImpact API コールの最大数
StartProtectedJob リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの StartProtectedJob API コールの最大数
StartProtectedQuery リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの StartProtectedQuery API 呼び出しの最大数
TagResource リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの TagResource API コールの最大数
UntagResource リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UntagResource API コールの最大数

名前	デフォルト	引き上げ可能	説明
UpdateAnalysisTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateAnalysisTemplate API コールの最大数
UpdateCollaboration リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateCollaboration API 呼び出しの最大数
UpdateCollaborationChangeRequest リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateCollaborationChangeRequest API コールの最大数
UpdateConfiguredAudienceModelAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateConfiguredAudienceModelAssociation API コールの最大数
UpdateConfiguredTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateConfiguredTable API 呼び出しの最大数
UpdateConfiguredTableAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateConfiguredTableAnalysisRule API 呼び出しの最大数
UpdateConfiguredTableAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateConfiguredTableAssociation API 呼び出しの最大数

名前	デフォルト	引き上げ可能	説明
UpdateConfiguredTableAssociationAnalysisRule リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateConfiguredTableAssociationAnalysisRule API コールの最大数
UpdateIdMappingTable リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateIdMappingTable API コールの最大数
UpdateIdNamespaceAssociation リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateIdNamespaceAssociation API コールの最大数
UpdatePrivacyBudgetTemplate リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdatePrivacyBudgetTemplate API コールの最大数
UpdateProtectedJob リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateProtectedJob API コールの最大数
UpdateProtectedQuery リクエストのレート	サポートされている各リージョン: 5/秒	あり	1 秒あたりの UpdateProtectedQuery API コールの最大数
メンバーシップあたりのテーブルの関連付け	サポートされている各リージョン: 25	可能	メンバーシップあたりのテーブルの関連付けの最大数

AWS Clean Rooms リソースパラメータの制限

[リソース]	デフォルト	説明
クエリテキストの長さ	90 KB	SQL クエリステートメントの最大テキスト長
クエリテキストの長さ (差分プライバシーを使用)	90 KB	差分プライバシーを使用する SQL クエリステートメントの最大テキスト長
クエリ実行時間	24 時間	タイムアウトまでのクエリの最大実行時間

AWS Clean Rooms ML クォータ

AWS アカウント には、AWS Clean Rooms ML の制限と呼ばれるデフォルトのクォータがあります。

AWS Clean Rooms ML のサービスクォータを表示するには、次のいずれかを実行します。

- Service Service Quotasユーザーガイドの [「サービスクォータの表示」](#) の手順に従って、ML AWS Clean Rooms をサービスとして選択します。
- の [「AWS Clean Rooms ML サービスクォータ」](#) を参照してくださいAmazon Web Services 全般のリファレンス。

サービスのパフォーマンスを維持し、AWS Clean Rooms ML の適切な使用を確保するために、アカウントに割り当てられたデフォルトのクォータは、リージョンの要因、支払い履歴、不正使用、および/またはクォータ引き上げリクエストの承認に応じて更新される場合があります。

名前	デフォルト	引き上げ可能	説明
オーディエンス生成ジョブあたりのアクティブなオーディエンスエクスポートジョブ	サポートされている各リージョン: 25	いいえ	オーディエンス生成ジョブのアクティブなオーディエンスエクスポートジョブの最大数
メンバーシップあたりのアクティブな設定済みモデルアルゴリズムの関連付け	サポートされている各リージョン: 1,000	あり	メンバーシップあたりのアクティブな設定済みモデルアルゴリズムの関連付けの最大数
メンバーシップあたりのアクティブな設定済みモデルアルゴリズム	サポートされている各リージョン: 1,000	あり	メンバーシップあたりのアクティブな設定済みモデルアルゴリズムの最大数
メンバーシップあたりのアクティブなカスタムモデルの入力チャネル	サポートされている各リージョン: 100	可能	メンバーシップあたりのアクティブなカスタムモデルの入力チャネルの最大数
アカウントあたりのアクティブなトレーニングインスタンスの最大数	サポートされている各リージョン: 10	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりのアクティブなトレーニングインスタンスの最大数。
トレーニング済みモデルあたりのアクティブ/保留中/進行中のトレーニング済みモデルバージョンの最大数	サポートされている各リージョン: 100	可能	トレーニング済みモデルあたりの作成可能なアクティブ/保留中/進行中の

名前	デフォルト	引き上げ可能	説明
			トレーニング済みモデルバージョンの最大数。
合成データを生成する入力列の最大数	サポートされている各リージョン: 1,000	いいえ	合成データを生成する入力列の最大数
合成データを生成する入力行の最大数	サポートされている各リージョン: 2,500,000	いいえ	合成データを生成する入力行の最大数
アカウントあたりの ml.c4.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c4.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c4.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c4.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c4.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c4.8xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c4.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c4.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5.18xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5.18xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c5.9xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5.9xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5n.18xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5n.18xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5n.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5n.2xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c5n.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5n.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5n.9xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5n.9xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c5n.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c5n.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c6i.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.12xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c6i.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.16xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c6i.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c6i.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c6i.32xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.32xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c6i.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c6i.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c6i.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c6i.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.12xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c7i.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.16xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.48xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.c7i.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.large トレーニングインスタンスの最大数。
アカウントあたりの ml.c7i.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.c7i.xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g4dn.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g4dn.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g4dn.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g4dn.16xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g4dn.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g4dn.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g4dn.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g4dn.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g4dn.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g4dn.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g4dn.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g4dn.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g5.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g5.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g5.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g5.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g5.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g5.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g5.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g5.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g5.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g6.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g6.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6e.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6e.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g6e.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6e.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6e.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6e.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.g6e.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.g6e.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.g6e.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m4.10xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m4.10xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m4.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m4.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m4.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m4.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m4.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m4.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m4.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m4.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m5.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 3	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m5.12xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m5.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m5.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m5.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 3	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m5.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m5.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 3	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m5.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m5.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m5.large トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m5.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m5.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.16xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.24xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m6i.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.32xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.32xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.8xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m6i.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.large トレーニングインスタンスの最大数。
アカウントあたりの ml.m6i.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m6i.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m7i.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.m7i.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.large トレーニングインスタンスの最大数。
アカウントあたりの ml.m7i.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.m7i.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.p2.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p2.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.p2.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p2.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.p2.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p2.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.p4d.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p4d.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.p4de.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p4de.24xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.p5.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p5.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.p5en.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.p5en.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.r5.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.8xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.r5.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.large トレーニングインスタンスの最大数。
アカウントあたりの ml.r5.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5d.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5d.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.r5d.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5d.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5d.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.4xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r5d.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.8xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.r5d.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.large トレーニングインスタンスの最大数。
アカウントあたりの ml.r5d.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r5d.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.12xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.12xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.16xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.16xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.r7i.24xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.24xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.2xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.48xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.48xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.4xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.4xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.r7i.8xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.8xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.large トレーニングインスタンスの最大数。
アカウントあたりの ml.r7i.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.r7i.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.t3.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.t3.2xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.t3.large トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.t3.large トレーニングインスタンスの最大数。
アカウントあたりの ml.t3.medium トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.t3.medium トレーニングインスタンスの最大数。
アカウントあたりの ml.t3.xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.t3.xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.trn1.2xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.trn1.2xlarge トレーニングインスタンスの最大数。

名前	デフォルト	引き上げ可能	説明
アカウントあたりの ml.trn1.32xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.trn1.32xlarge トレーニングインスタンスの最大数。
アカウントあたりの ml.trn1n.32xlarge トレーニングインスタンスの最大数	サポートされている各リージョン: 0	あり	トレーニング済みモデルを作成するために使用できるアカウントあたりの ml.trn1n.32xlarge トレーニングインスタンスの最大数。
カテゴリ予測列の最大一意値	サポートされている各リージョン: 100	いいえ	カテゴリ予測列の最大一意値
合成データの生成に必要な入力列の最小数	サポートされている各リージョン: 5	いいえ	合成データの生成に必要な入力列の最小数
合成データの生成に必要な入力行の最小数	サポートされている各リージョン: 1,500	いいえ	合成データの生成に必要な入力行の最小数
カスタマーあたりの保留中/進行中のオーディエンスエクスポートジョブ	サポートされている各リージョン: 20	いいえ	カスタマーあたりの保留中/進行中のオーディエンスエクスポートジョブの最大数

名前	デフォルト	引き上げ可能	説明
カスタマーあたりの保留中/進行中のオーディエンス生成ジョブ	サポートされている各リージョン: 10	あり	カスタマーあたりの保留中/進行中のオーディエンス生成ジョブの最大数
カスタマーあたりの保留中/進行中のオーディエンスモデル	サポートされている各リージョン: 2	あり	カスタマーあたりの保留中/進行中のオーディエンスモデルトレーニングジョブの最大数
アカウントあたりの保留中/進行中のカスタムモデル推論ジョブ	サポートされている各リージョン: 10	あり	アカウントあたりの保留中/進行中のカスタムモデル推論ジョブの最大数
メンバーシップあたりの保留中/進行中のカスタムモデル推論ジョブ	サポートされている各リージョン: 5	あり	メンバーシップあたりの保留中/進行中のカスタムモデル推論ジョブの最大数
アカウントあたりの保留中/進行中のカスタムモデルトレーニングジョブ	サポートされている各リージョン: 10	あり	アカウントあたりの保留中/進行中のカスタムモデルトレーニングジョブの最大数
メンバーシップあたりの保留中/進行中のカスタムモデルトレーニングジョブ	サポートされている各リージョン: 5	あり	メンバーシップあたりの保留中/進行中のカスタムモデルトレーニングジョブの最大数
アカウントあたりの保留中/進行中の合成 MLIC 生成ジョブ	サポートされている各リージョン: 2	あり	アカウントあたりの保留中/進行中の合成 MLIC 生成ジョブの最大数

AWS Clean Rooms ML には、以下に示す追加のクォータがあります。

Clean Rooms ML のクォータ

[リソース]	デフォルト	説明
カスタムモデリング推論ジョブの最大期間	25 時間	
インタラクションの最大数	200 億	トレーニングデータ内のインタラクションの最大数。より大きな入力サンプルはサンプルダウンされます。
インタラクションの最小数	100 万件	
類似モデルトレーニングの個別ユーザーの最大数	1 億件	さらに多く含まれる場合は、インタラクションの数でランク付けされた上位 1 億件のみが使用されます。
類似モデルのトレーニングに必要な個別ユーザーの最小数	100,000	
エクスポート類似セグメント (オーディエンス) ジョブの最小ユーザー数	10,000	
モデルトレーニングに使用される個別アイテムの最大数。	100 万件	最大 5,000 万項目まで含めることができますが、使用されるのは最も一般的な 100 万項目だけです。
トレーニングデータセット内の特徴量列の最大数。	10	
ユーザーあたりの個別の項目の最小数	2	AWS Clean Rooms ML では、各行またはユーザーに、繰り返し項目を含む 2 つ以上の項目が必要です。

[リソース]	デフォルト	説明
シードオーディエンスの最大サイズ	500,000	
シードオーディエンスの最小サイズ	500	トレーニングデータプロバイダーは、この値を最低 25 に設定できます。
API	お客様 1 人あたり	
アクティブなトレーニングデータセットの総数。	500	
アクティブな類似モデル (オーディエンスモデル) の総数	500	
設定済みのアクティブな類似モデル (オーディエンスモデル) の総数	10,000	
完了した類似セグメント (オーディエンス) 生成ジョブの総数	無制限	
エクスポートが完了した類似セグメント (オーディエンス) ジョブの総数	無制限	
類似モデル (オーディエンスモデル) 生成ジョブの最大所要時間	1 日 (24 時間)	

[リソース]	デフォルト	説明
類似セグメント (オーディエンス) 生成ジョブの最大所要時間	10 時間	シードを指定した後で、Clean Rooms ML が類似セグメントを生成するまでに最大 10 時間かかります。SQL クエリをシードデータとして使用すると、類似セグメントの生成にかかる 10 時間に加えて、クエリの実行に最大 12 時間かかります。
セグメント (オーディエンス) サイズのビンの最小割合	1%	
セグメント (オーディエンス) サイズのビンの最大割合	20%	
セグメント (オーディエンス) サイズのビンの最小絶対サイズ	個別ユーザー数の 1%	
セグメント (オーディエンス) サイズのビンの最大絶対サイズ	個別ユーザー数の 20%	

Clean Rooms ML API スロットリングクォータ

には、エンドポイントクォータごとにアカウントごとに次の API トランザクション/秒 (TPS) AWS アカウント があります。

[リソース]	[Rate limit] (レート制限)	説明
CreateAudienceModel リクエストのレート	1 TPS レート、3 TPS バースト	1 秒あたりの CreateAudienceModel API コールの最大数

[リソース]	[Rate limit] (レート制限)	説明
CreateConfiguredAudienceModel リクエストのレート	10 TPS	1 秒あたりの CreateConfiguredAudienceModel API コールの最大数
CreateTrainingDataset リクエストのレート	10 TPS	1 秒あたりの CreateTrainingDataset API コールの最大数
DeleteAudienceGenerationJob リクエストのレート	2 TPSレート、10TPSバースト	1 秒あたりの DeleteAudienceGenerationJob API コールの最大数
DeleteAudienceModel リクエストのレート	2 TPSレート、10TPSバースト	1 秒あたりの DeleteAudienceModel API コールの最大数
DeleteConfiguredAudienceModel リクエストのレート	10 TPS	1 秒あたりの DeleteConfiguredAudienceModel API コールの最大数
DeleteConfiguredAudienceModelPolicy リクエストのレート	25 TPS	1 秒あたりの DeleteConfiguredAudienceModelPolicy API コールの最大数
DeleteTrainingDataset リクエストのレート	10 TPS	1 秒あたりの DeleteTrainingDataset API コールの最大数
GetAudienceGenerationJob リクエストのレート	50 TPS	1 秒あたりの GetAudienceGenerationJob API コールの最大数
GetAudienceModel リクエストのレート	50 TPS	1 秒あたりの GetAudienceModel API コールの最大数

[リソース]	[Rate limit] (レート制限)	説明
GetConfiguredAudienceModel リクエストのレート	50 TPS	1 秒あたりの GetConfiguredAudienceModel API コールの最大数
GetConfiguredAudienceModelPolicy リクエストのレート	50 TPS	1 秒あたりの GetConfiguredAudienceModelPolicy API コールの最大数
GetTrainingDataset リクエストのレート	50 TPS	1 秒あたりの GetTrainingDataset API コールの最大数
ListAudienceExportJobs リクエストのレート	50 TPS	1 秒あたりの ListAudienceExportJobs API コールの最大数
ListAudienceGenerationJobs リクエストのレート	50 TPS	1 秒あたりの ListAudienceGenerationJobs API コールの最大数
ListAudienceModels リクエストのレート	50 TPS	1 秒あたりの ListAudienceModels API コールの最大数
ListConfiguredAudienceModels リクエストのレート	50 TPS	1 秒あたりの ListConfiguredAudienceModels API コールの最大数
ListTagsForResource リクエストのレート	50 TPS	1 秒あたりの ListTagsForResource API コールの最大数
ListTrainingDatasets リクエストのレート	50 TPS	1 秒あたりの ListTrainingDatasets API コールの最大数

[リソース]	[Rate limit] (レート制限)	説明
PutConfiguredAudienceModelPolicy リクエストのレート	25 TPS	1 秒あたりの PutConfiguredAudienceModelPolicy API コールの最大数
StartAudienceExportJob リクエストのレート	1 TPS レート、3 TPS バースト	1 秒あたりの StartAudienceExportJob API コールの最大数
StartAudienceGenerationJob リクエストのレート	1 TPS レート、5 TPS バースト	1 秒あたりの StartAudienceGenerationJob API コールの最大数
TagResource リクエストのレート	10 TPS	1 秒あたりの TagResource API コールの最大数
UntagResource リクエストのレート	50 TPS	1 秒あたりの UntagResource API コールの最大数
UpdateConfiguredAudienceModel リクエストのレート	10 TPS	1 秒あたりの UpdateConfiguredAudienceModel API コールの最大数
CreateConfiguredModelAlgorithm リクエストのレート	10 TPS	1 秒あたりの CreateConfiguredModelAlgorithm API コールの最大数。
CreateConfiguredModelAlgorithmAssociation リクエストのレート	10 TPS	1 秒あたりの CreateConfiguredModelAlgorithmAssociation API コールの最大数。
PutMLConfiguration リクエストのレート	10 TPS	1 秒あたりの PutMLConfiguration API コールの最大数。

[リソース]	[Rate limit] (レート制限)	説明
CreateTrainedModel リクエストのレート	1 TPS レート、3 TPS バースト	1 秒あたりの CreateTrainedModel API コールの最大数。
StartTrainedModelExportJob リクエストのレート	10 TPS	1 秒あたりの StartTrainedModelExportJob API コールの最大数。
StartTrainedModelInferenceJob リクエストのレート	1 TPS レート、3 TPS レート	1 秒あたりの StartTrainedModelInferenceJob API コールの最大数。
GetConfiguredModelAlgorithm リクエストのレート	50 TPS	1 秒あたりの GetConfiguredModelAlgorithm API コールの最大数。
GetConfiguredModelAlgorithmAssociation リクエストのレート	50 TPS	1 秒あたりの GetConfiguredModelAlgorithmAssociation API コールの最大数。
GetTrainedModel リクエストのレート	50 TPS	1 秒あたりの GetTrainedModel API コールの最大数。
GetMLConfiguration リクエストのレート	50 TPS	1 秒あたりの GetMLConfiguration API コールの最大数。
GetTrainedModelInferenceJob リクエストのレート	50 TPS	1 秒あたりの GetTrainedModelInferenceJob API コールの最大数。
ListConfiguredModelAlgorithm リクエストのレート	50 TPS	1 秒あたりの ListConfiguredModelAlgorithm API コールの最大数。

[リソース]	[Rate limit] (レート制限)	説明
ListConfiguredModelAlgorithmAssociations リクエストのレート	50 TPS	1 秒あたりの ListConfiguredModelAlgorithmAssociations API コールの最大数。
ListTrainedModels リクエストのレート	50 TPS	1 秒あたりの ListTrainedModels API コールの最大数。
ListCollaborationTrainedModelExportJobs リクエストのレート	50 TPS	1 秒あたりの ListCollaborationTrainedModelExportJobs API コールの最大数。
ListCollaborationTrainedModelInferenceJobs リクエストのレート	50 TPS	1 秒あたりの ListCollaborationTrainedModelInferenceJobs API コールの最大数。
DeleteConfiguredModelAlgorithm リクエストのレート	2 TPSレート、10TPSバースト	1 秒あたりの DeleteConfiguredModelAlgorithm API コールの最大数。
DeleteConfiguredModelAlgorithmAssociation リクエストのレート	2 TPSレート、10TPSバースト	1 秒あたりの DeleteConfiguredModelAlgorithmAssociation API リクエストの最大数。
DeleteMLConfiguration リクエストのレート	2 TPSレート、10TPSバースト	1 秒あたりの DeleteMLConfiguration API リクエストの最大数。
DeleteTrainedModelOutput リクエストのレート	2 TPSレート、10TPSバースト	1 秒あたりの DeleteTrainedModelOutput API リクエストの最大数。

AWS Clean Rooms ユーザーガイドのドキュメント履歴

次の表に、のドキュメントリリースを示します AWS Clean Rooms。

このドキュメントの更新に関する通知については、RSS フィードにサブスクライブできます。RSS の更新をサブスクリプションするには、使用しているブラウザで RSS プラグインを有効にする必要があります。

変更	説明	日付
コスト配分タグ付けセクションのドキュメントを更新	コスト配分関連のタグ付け可能なリソースに関するドキュメントを追加 AWS Billing and Cost Management し、セクションを作成しました。	2026 年 2 月 20 日
cleanrooms-ml spark コンピューティング設定のサポート	SQL クエリでサポートされている Spark プロパティを設定することで、Spark ランタイムの動作をカスタマイズし、SQL パフォーマンスを最適化できるようになりました。この機能は、 オーディエンス生成ジョブの作成 と MLInputChannel の作成 で使用できるようになりました。	2026 年 1 月 5 日
CloudWatch による詳細モニタリングのサポート	お客様は、クエリのパフォーマンスやリソース使用率など、コラボレーションの運用モニタリングのための CloudWatch メトリクスによる詳細モニタリングにオプションできるようになりました。	2026 年 1 月 2 日

[コラボレーション変更リクエスト EventBridge 通知イベントのサポート](#)

コラボレーション変更リクエストのライフサイクルイベントの新しい EventBridge イベントを追加しました。

2025 年 12 月 19 日

[コラボレーションメンバーシップと設定を管理するための変更リクエストのサポート](#)

コラボレーションクリエイターは、変更リクエストを送信して、新しいメンバーの追加、既存のメンバー機能の更新、自動承認設定を変更できるようになりました。すべてのコラボレーションメンバーは、変更が有効になる前に変更リクエストを承認する必要があります。

2025 年 12 月 18 日

[AWS Clean Rooms SQL 分析エンジンのサポート終了タイムライン](#)

AWS Clean Rooms SQL 分析エンジンのサポートは 2025 年 12 月 17 日に終了します。お客様は Spark SQL 分析エンジンのみを使用できるようになりました。

2025 年 12 月 17 日

[追加の EventBridge 通知イベントのサポート](#)

分析テンプレート、設定済みテーブルの関連付け、設定済みテーブルの関連付け分析ルール、ID マッピングテーブル、ID 名前空間の関連付け、保護済みジョブ、保護済みクエリの新しい EventBridge イベントを追加しました。

2025 年 12 月 15 日

[カスタム機械学習モデルをトレーニングするためのプライバシー強化合成データセット生成のサポート](#)

お客様は、プライバシーが強化された合成データセットを生成して、機密データを保護しながらカスタム機械学習モデルをトレーニングできるようになりました。

2025 年 11 月 30 日

[SQL クエリで設定可能な Spark プロパティのサポート](#)

SQL クエリでサポートされている Spark プロパティを設定することで、Spark ランタイムの動作をカスタマイズできるようになりました。

2025 年 10 月 30 日

[クロスリージョンコラボレーションのサポート](#)

AWS Clean Rooms は、AWS Glue と Athena データテーブルのクロスリージョンコラボレーションをサポートするようになりました。これにより、お客様は 1 つのリージョンに保存されているデータを処理 AWS リージョンして、すべてのコラボレーションメンバーによって承認されたリージョンに結果を配信できます。

2025 年 10 月 3 日

[データアクセス予算のサポート](#)

お客様は、データアクセス予算を設定および管理して、テーブルの使用状況を制御できるようになりました。

2025 年 10 月 1 日

[ID マッピングテーブルの入力 – 更新](#)

お客様は、ルールベースの ID マッピングワークフローで増分処理を使用して、大規模なデータセットをより効率的に処理できるようになりました。お客様は、ID マッピングワークフローからレコードを削除して、データ管理規制に準拠することもできます。

2025 年 9 月 23 日

[PySpark ジョブの可変ワーク ロードサイズをサポート](#)

お客様は、コンピューティングサイズを選択して調整し、PySpark ジョブをより効果的に実行できるようになりました。

2025 年 9 月 4 日

[既存のコラボレーションに新 しいメンバーを追加するた めのサポート](#)

お客様は既存のコラボレーションに新しいメンバーを追加できるようになりました。この機能には、自動承認設定とブロックメンバーシップリクエストを設定するオプションが含まれています。

2025 年 9 月 3 日

[カスタムモデルログのエラー 概要とリダクションの設定 のサポートを追加](#)

お客様は、設定されたモデルアルゴリズムの関連付けで、カスタムモデルログのエラー概要とリダクションを設定できるようになりました。

2025 年 9 月 3 日

[Snowflake 認証オプションの サポート](#)

接続 AWS Secrets Manager のために Snowflake 認証情報を保存するときに、Snowflake PEM プライベートキーでキーペア認証を使用できるようになりました AWS Clean Rooms。

2025 年 8 月 29 日

[PySpark 分析テンプレートのエラー設定を追加](#)

お客様は、PySpark 分析テンプレートで基本的なエラーメッセージレベルまたは詳細なエラーメッセージレベルを設定して、エラーの可視性とトラブルシューティング機能を管理できるようになりました。

2025 年 8 月 19 日

[EventBridge のサポート](#)

お客様は Amazon EventBridge を使用して、から他のソフトウェアコンポーネントにイベント AWS Clean Rooms をルーティングできるようになりました。

2025 年 7 月 31 日

[既存のポリシーの更新](#)

AWSCleanRoomsFullAccessNoQuerying 管理ポリシーに次の新しいアクセス許可が追加されました: `cleanrooms:UpdateConfiguredTableAllowedColumns` および `cleanrooms:UpdateConfiguredTableReference`。

2025 年 7 月 29 日

[設定済みテーブルの編集機能の強化](#)

お客様は、設定済みテーブルの許可された列とテーブル参照フィールドを変更できるようになりました。

2025 年 7 月 29 日

[ML 入力チャネルが Parquet ファイルタイプをサポート](#)

お客様は、ML 入力チャネルが使用できるデータ形式として Parquet ファイルタイプを選択できるようになりました。

2025 年 7 月 17 日

[Spark SQL 分析エンジンの差分プライバシーのサポート](#)

お客様は、Spark SQL 分析エンジンで差分プライバシーを使用できるようになりました。さらに、差分プライバシーを使用する SQL クエリステートメントの最大テキスト長は 90 KB になりました。

2025 年 7 月 16 日

[AWS Clean Rooms SQL 分析エンジンのサポート終了タイムライン](#)

Clean Rooms SQL 分析エンジンを使用した新しいコラボレーションのサポートは、2025 年 7 月 16 日に終了します。2025 年 7 月 17 日現在、従来の Clean Rooms SQL 分析エンジンとの新しいコラボレーションを作成することはできません。は、2025 年 12 月 17 日に Clean Rooms SQL 分析エンジンのサポート AWS Clean Rooms を終了します。

2025 年 7 月 16 日

[増分トレーニングと分散トレーニングのサポート](#)

お客様は、カスタムモデルに増分トレーニングと分散トレーニングを使用できるようになりました。

2025 年 7 月 1 日

[Clean Rooms SQL 分析エンジンのサポート終了タイムライン](#)

2025 年 7 月 16 日まで、従来の Clean Rooms SQL 分析エンジンを使用した新しいコラボレーションには制限の引き上げが必要です。2025 年 7 月 17 日以降、このエンジンとの新しいコラボレーションを作成することはできません。は、2025 年 12 月 17 日に Clean Rooms SQL 分析エンジンのサポート AWS Clean Rooms を終了します。

2025 年 5 月 22 日

[PySpark 分析テンプレートの更新](#)

お客様は、任意の有効な Python ファイル名でユーザースクリプトファイルに名前を付けることができるようになりました。

2025 年 5 月 15 日

[複数の結果レシーバーのサポート](#)

お客様は複数のメンバーを選択して、1 つのクエリからクエリ結果を受信できるようになりました。

2025 年 4 月 30 日

[Spark SQL へのコラボレーションの移行のサポート](#)

AWS Clean Rooms SQL は、カスタム分析ルールに加えて、集約およびリスト分析ルールをサポートするようになりました。さらに、お客様は既存のコラボレーションを更新して、Spark SQL を強化する Spark 分析エンジンを使用できます。

2025 年 4 月 2 日

[PySpark ジョブのサポート](#)

お客様は、承認された PySpark 分析テンプレートを使用してジョブを実行してデータを分析できるようになりました。

2025 年 3 月 18 日

[既存のポリシーに対する更新](#)

次の新しい権限が AWSCleanRoomsMLReadOnlyAccess マネージドポリシーに追加されました: PassCleanRoomsResources AWSCleanRoomsMLFullAccess 管理ポリシーに次の新しいアクセス許可が追加されました: PassCleanRoomsResources および ConsoleDescribeECRRepositories。

2025 年 1 月 10 日

[複数のコンピューティングワーカーのサポート](#)

お客様は、類似セグメントの作成時にプロビジョニングするコンピューティングワーカーのタイプと数を指定できるようになりました。

2024 年 12 月 17 日

[複数のデータソースとクラウドのサポート](#)

お客様は、Amazon Athena や Snowflake などの複数のデータソースとクラウドを使用して、パートナーのデータセットとコラボレーションできるようになりました。

2024 年 12 月 1 日

[Clean Rooms ML Custom Modeling が利用可能になりました](#)

お客様は、コラボレーションで独自のカスタム ML モデルを使用できるようになりました。

2024 年 11 月 7 日

[新しい分析エンジン](#)

大規模なデータセットを持つお客様は、Spark SQL 分析エンジンでサポートされている SQL 関数を使用して複雑なクエリを実行できるようになりました。

2024 年 10 月 29 日

[プライバシー保護の強化、類似オーディエンスの構築、複数の結果レシーバーの選択](#)

[追加分析] と [コラボレーション分析ルール] を使用して、複雑なアクティベーションクエリを許可しながらデータを保護できます。SQL クエリまたは分析テンプレートから類似オーディエンスモデルを構築できます。複数のメンバーを選択して結果を受け取ることができます。

2024 年 7 月 24 日

[でのエンティティ解決 AWS Clean Rooms](#)

in を使用すると AWS Entity Resolution AWS Clean Rooms、2 つの ID 名前空間間に ID マッピングテーブルを作成して、異なる ID 空間間でイベントデータをクエリできます。

2024 年 7 月 23 日

[既存のポリシーの更新](#)

次の新しい権限がAWSCleanRoomsFullAccessNoQuerying マネージドポリシーに追加されました:cleanrooms:BatchGetSchemaAnalysisRule

2024 年 5 月 13 日

[AWS Clean Rooms ML が完全に利用可能になりました](#)

AWS Clean Rooms ML は、2 つの当事者間でデータを共有しなくても、データ内の類似ユーザーを識別するためのプライバシー強化方法を提供します。

2024 年 4 月 3 日

[既存のポリシーの更新](#)

AWSCleanRoomsFullAccess マネージドポリシーのステートメント ID が ConsolePickQueryResultsBucket から SetQueryResultsBucket に更新され、アクセス許可降のアクセス許可をより適切に表すことができるようになりました。

2024 年 3 月 21 日

[AWS Clean Rooms ML の新しい管理ポリシー](#)

AWSCleanRoomsMLReadOnlyAccess と AWSCleanRoomsMLFullAccess という 2 つの新しいマネージドポリシーが追加されました。

2023 年 11 月 29 日

[AWS Clean Rooms ML \(プレビュー\)](#)

AWS Clean Rooms ML は、2 つの当事者間でデータを共有しなくても、データ内の類似ユーザーを識別するためのプライバシー強化方法を提供します。

2023 年 11 月 29 日

[AWS Clean Rooms 差分プライバシー \(プレビュー\)](#)

お客様は、AWS Clean Rooms 差分プライバシーを使用してユーザーのプライバシーを保護することができます。

2023 年 11 月 29 日

[支払い設定](#)

コラボレーションクリエーターが、クエリを行えるメンバーまたはコラボレーション内の別のメンバーを、クエリの計算コストの請求先として設定できるようになりました。

2023 年 11 月 14 日

[クエリ実行時間 - 更新](#)

クエリ実行時間のタイムアウトまでの上限が 4 時間から 12 時間に更新されました。

2023 年 10 月 6 日

[CloudFormation リソース - 更新](#)

AWS Clean Rooms は、`AWS::CleanRooms::MembershipProtectedQueryOutputConfiguration`、`AWS::CleanRooms::MembershipProtectedQueryResultConfiguration` およびの新しいリソースを追加しました`AWS::CleanRooms::MembershipProtectedQueryS3OutputConfiguration`。

2023 年 9 月 7 日

[CloudFormation リソース - 更新](#)

AWS Clean Rooms では、`AWS::CleanRooms::AnalysisTemplate` という新しいリソースが追加されました`AWS::CleanRooms::ConfiguredTableAnalysisRuleCustom`。

2023 年 8 月 31 日

[メンバー能力の個別指定](#)

コラボレーションクリエイターが、クエリを行えるメンバーを1人と、結果を受け取れる別のメンバーを1人指定できるようになりました。これにより、コラボレーションクリエイターは、クエリを実行するメンバーがクエリ結果にアクセスできないように設定できます。

2023 年 8 月 30 日

[AWS Clean Rooms 用語集](#)

AWS Clean Rooms 用語集を追加するためのドキュメントのみの更新。

2023 年 8 月 30 日

[Apache Iceberg テーブルのサポート \(プレビュー\)](#)

AWS Clean Rooms でApache Icebergテーブルがサポートされるようになりました (プレビュー)。

2023 年 8 月 25 日

[クォータの更新](#)

アカウントごとのメンバーシップの新しいデフォルトクォータを反映して、[「クォータ」セクション](#)が更新されました。

2023 年 8 月 9 日

既存のポリシーの更新

2023 年 7 月 31 日

AWSCleanRoomsFullAccessNoQuerying マネージドポリシーに次の新しいアクセス許可が追加されました。cleanrooms:CreateAnalysisTemplate、cleanrooms:GetAnalysisTemplate、cleanrooms:UpdateAnalysisTemplate、cleanrooms>DeleteAnalysisTemplate、cleanrooms>ListAnalysisTemplates、cleanrooms:GetCollaborationAnalysisTemplate、cleanrooms:BatchGetCollaborationAnalysisTemplate、および cleanrooms>ListCollaborationAnalysisTemplates。

[分析テンプレートとカスタム分析ルール](#)

AWS Clean Rooms は、分析テンプレートとカスタム分析ルールをサポートするようになりました。分析テンプレートにより、コラボレーターは独自のカスタム SQL クエリを作成またはインポートしてコラボレーションで使用できます。カスタム分析ルールを使用すると、テーブル所有者は設定済みテーブルに対するカスタム SQL クエリを承認できます。

2023 年 7 月 31 日

[分析ルールでの OR 論理条件のサポート](#)

AWS Clean Rooms 分析ルールが OR JOIN 句の論理条件をサポートするようになりました。

2023 年 6 月 29 日

[CloudFormation との統合](#)

AWS Clean Rooms がと統合されるようになりました CloudFormation。

2023 年 6 月 15 日

[分析ビルダー](#)

クエリを実行して結果を受け取ることができるメンバーは、[分析ビルダー UI] を使用して、SQL コードを記述することなく一部のテーブルでクエリを実行できるようになりました。

2023 年 6 月 15 日

[SQL 関数](#)

ドキュメントのみの更新。サポートされている SQL 関数を明確にしました。

2023 年 5 月 5 日

トラブルシューティング	ドキュメントのみの更新。一般的な問題に関するトラブルシューティングセクションを追加しました。	2023 年 4 月 27 日
でサポートされているデータ型 AWS Clean Rooms	サポートされている AWS Glue Data Catalog データ型を一覧表示する新しいセクションを追加するためのドキュメントのみの更新。	2023 年 4 月 26 日
AWS CloudTrail イベントの例	ドキュメントのみの更新。CloudTrail イベントの StartProtectedQuery (成功) と StartProtectedQuery (失敗) の例を追加しました。	2023 年 4 月 20 日
既存のポリシーの更新	AWSCleanRoomsFullAccessNoQuerying マネージドポリシーに次の新しいアクセス許可が追加されました。cleanrooms:ListTagsForResource、cleanrooms:UntagResource、および cleanrooms:TagResource。詳細については、「 AWS マネージドポリシー 」を参照してください。	2023 年 3 月 21 日
一般提供	AWS Clean Rooms が一般公開されました。	2023 年 3 月 21 日
プレビューリリース	AWS Clean Rooms ユーザーガイドのプレビューリリース	2023 年 1 月 12 日

AWS Clean Rooms 用語集

この用語集を参照して、AWS Clean Roomsで使用される用語を理解してください。

集計分析ルール

COUNT、SUM、または AVG 関数を使用して任意のディメンションで分析を集約するクエリを許可するクエリ制限です。このようなクエリでは行レベルの情報は明らかにはなりません。

キャンペーン計画、メディアリーチ、頻度測定、コンバージョン測定などのユースケースに対応します。

その他の分析ルールタイプとして、[カスタム](#)と[リスト](#)があります。

分析ルール

特定のタイプのクエリを許可するクエリ制限です。

分析ルールのタイプによって、設定済みテーブルに対して実行できる分析の種類が決まります。タイプごとに事前定義されたクエリ構造があります。テーブル列を構造内でどのように使用するかは、クエリコントロールを通じて制御します。

分析ルールには、[集計](#)、[リスト](#)、[カスタム](#)の 3 種類があります。

分析テンプレート

コラボレーションに固有の、事前に承認された、再利用可能なクエリです。

サポートされている形式: Spark の SQL コードまたは Python コード。

SQL を使用する場合、分析テンプレートには、通常 SQL クエリにリテラル値が表示される可能性のあるパラメータを含めることができます。サポートされているパラメータタイプの詳細については、「AWS Clean Rooms SQL リファレンス」の「[データ型](#)」を参照してください。

分析テンプレートは[カスタム分析ルール](#)でのみ使用できます。

C3R 暗号化クライアント

Cryptographic Computing for Clean Rooms (C3R) の暗号化クライアントです。

データの暗号化と復号化に使用される C3R は、コマンドラインインターフェイスを備えたクライアント側の暗号化 SDK です。

クリアテキスト列

JOIN または SELECT SQL コンストラクトのどちらに対しても暗号化によって保護されていない列です。

クリアテキスト列は SQL クエリのどの部分でも使用できます。

コラボレーション

メンバーが設定済みテーブル AWS Clean Rooms に対して SQL クエリを実行できる安全な論理境界。

コラボレーションは[コラボレーションクリエイター](#)が作成します。

コラボレーションに参加できるのは、コラボレーションに招待されたメンバーだけです。

コラボレーションには、データを[クエリできるメンバー](#)を 1 人のみ持つことも、[クエリとジョブを実行できるメンバーを 1 人](#)持つこともできます。

コラボレーションには、[結果を受け取ることができるメンバーを 1 人](#)のみ持つことができます。

コラボレーションでは、[クエリコンピューティングコストを支払うメンバー](#)を 1 人だけ、または[クエリとジョブコンピューティングコストを支払うメンバー](#)を 1 人だけ持つことができます。

すべてのメンバーは、コラボレーションに参加する前に、コラボレーションに招待された参加者のリストを確認できます。

コラボレーションクリエイター

コラボレーションを作成するメンバーです。

コラボレーションクリエイターは 1 つのコラボレーションにつき 1 人だけです。

コラボレーションからメンバーを削除したり、コラボレーションを削除したりできるのは、コラボレーションクリエイターだけです。

設定済みテーブル

各設定済みテーブルは、で使用するように AWS Glue Data Catalog 設定された 内の既存のテーブルへの参照を表します AWS Clean Rooms。設定済みテーブルには、データの使用方法を決定する分析ルールが含まれています。

現在、は、カタログ化された Amazon Simple Storage Service (Amazon S3) に保存されているデータの関連付け AWS Clean Rooms をサポートしています AWS Glue。

詳細については AWS Glue、「[AWS Glue デベロッパーガイド](#)」を参照してください。

設定済みテーブルは 1 つ以上のコラボレーションに関連付けることができます。

Note

AWS Clean Rooms は現在、に登録されている Amazon S3 バケットの場所をサポートしていません AWS Lake Formation。

カスタム分析ルール

事前に承認された特定のクエリセット ([分析テンプレート](#)) を許可するクエリ制限、またはデータを使用するクエリやジョブを提供できる特定のアカウントのセットを許可するクエリ制限。

ファーストタッチアトリビューション、増分分析、オーディエンス発掘分析などのユースケースに対応します。

差分プライバシーをサポートします。

その他の分析ルールには、[集計](#)と[リスト](#)があります。

復号

暗号化されたデータを元の形式に戻すプロセスです。復号化は、シークレットキーにアクセスできる場合にのみ実行できます。

差分プライバシー

特定の個人について学習した結果を受け取ることができるメンバーからコラボレーションデータを保護する、数学的に厳格な手法。

暗号化

キーと呼ばれる秘密の値を使用して、データをランダムに見える形式にエンコードするプロセスです。キーにアクセスしない限り、元のプレーンテキストを特定することはできません。

フィンガープリント列

JOIN SQL 構文のために暗号化によって保護されている列です。

ID マッピングワークフローメソッド

ID マッピングの実行方法。

ID マッピングワークフローには 2 つの方法があります。

- ルールベースの ID マッピング – 一致するルールを使用して、ID マッピングワークフローのソースからターゲットにファーストパーティータータを変換する方法。
- プロバイダーサービス ID マッピング – プロバイダーサービスを使用して、ID マッピングワークフローのソースからターゲットにサードパーティータータがエンコードしたデータを変換する方法。

AWS Clean Rooms は現在、プロバイダーのサービスベースの ID マッピングワークフローメソッドとして LiveRamp をサポートしています。この方法 AWS Data Exchange を使用するには、を通じて LiveRamp へのサブスクリプションが必要です。詳細については、「AWS Entity Resolution ユーザーガイド」の「[AWS Data Exchangeでプロバイダーサービスをサブスクライブする](#)」を参照してください。

ID マッピングテーブル

コラボレーションでファーストパーティマッチングルールまたはマルチパーティ ID トランスコード AWS Clean Rooms を有効にする のリソース。

ID マッピングテーブルは、AWS Glue Data Catalog内の既存のテーブルを参照するものです。これには、AWS Clean Roomsでどのようにデータにクエリを実行するかを決定する [ID マッピングテーブル分析ルール](#)が含まれています。ID マッピングテーブルは 1 つ以上のコラボレーションに関連付けることができます。

ID マッピングテーブル分析ルール

AWS Clean Rooms によって管理され、クエリを容易にするために異なる ID データを結合するために使用される分析ルールのタイプ。[ID マッピングテーブル](#)に自動的に追加され、編集することはできません。これは、それらの分析ルールが同種である限り、コラボレーション内の他の分析ルールの動作を継承します。

ID マッピングワークフロー

指定された [ID マッピングワークフローメソッド](#)に基づいて、ソースからターゲットにデータをマッピングするデータ処理ジョブ。これにより、ID [マッピングテーブル](#)が生成されます。

ID 名前空間

複数のデータセットを説明するメタデータ AWS アカウント と、[ID マッピングワークフロー](#)でこれらのデータセットを使用する方法 AWS Clean Rooms を含む のリソース。

ID 名前空間の関連付け

[ID マッピングワークフロー](#)への入力を検出するのに役立つ、ID 名前空間リソースの関連付け。

ジョブ

サポートされている関数、クラス、変数のセットを使用して、コラボレーション内の設定済みテーブルにアクセスして分析する方法。

AWS Clean Rooms は現在、PySpark ジョブタイプをサポートしています。

AWS Clean Rooms は現在、PySpark 分析テンプレートを使用したジョブの実行をサポートしています。

リスト分析ルール

クエリを行えるメンバーのテーブルとこのテーブルとの重複部分について、行レベルの属性分析を出力するクエリを許可するクエリ制限です。

エンリッチメント、オーディエンス構築、サプレッションなどのユースケースに対応します。

その他の分析ルールには、[集計](#)と[カスタム](#)があります。

類似モデル

シードデータプロバイダーが、シードデータに最も近いトレーニングデータプロバイダーのデータの[類似セグメント](#)を作成できるようにするトレーニングデータプロバイダーのデータのモデル。

[シードデータ](#)

類似セグメント

[シード](#)データに最も近いトレーニングデータのサブセット。

メンバー

[コラボレーション](#)に参加している AWS お客様。

メンバーは、それぞれの AWS アカウントを使用して識別されます。

メンバー全員がデータを寄稿できます。

クエリを行えるメンバー

[コラボレーション](#)においてデータにクエリを実行できるメンバーです。

クエリを行えるメンバーはコラボレーションごとに 1 人だけであり、そのメンバーは変更できません。

管理ユーザーは AWS Identity and Access Management (IAM) アクセス許可を使用して、コラボレーション内のデータをクエリできる IAM プリンシパル (ユーザーやロールなど) を制御できます。詳細については、「[Amazon S3 からデータを読み取るサービスロールを作成する](#)」を参照してください。

クエリとジョブを実行できるメンバー

[コラボレーション](#)内のデータに対してクエリとジョブを実行できるメンバー。

コラボレーションごとにクエリとジョブを実行できるメンバーは 1 人のみで、そのメンバーはイミュータブルです。

管理ユーザーは AWS Identity and Access Management (IAM) アクセス許可を使用して、コラボレーションでクエリやジョブを実行できる IAM プリンシパル (ユーザーやロールなど) を制御できます。詳細については、「[Amazon S3 からデータを読み取るサービスロールを作成する](#)」を参照してください。

結果を受け取れるメンバー

クエリ結果を受信できるメンバー。結果を受け取ることができるメンバーは、Amazon S3 の送信先とクエリ結果形式 (CSV または Parquet) のクエリ結果設定を指定します。

Spark 分析エンジンを使用した分析の場合、結果を受け取ることができるメンバーは、ファイルを 1 つのファイルとして出力するか、複数のファイルとして出力するかも指定します。

コラボレーションで結果を受け取ることができるメンバーは複数存在できます。

クエリの計算コストを負担するメンバー

クエリの計算コストの支払いを担当するメンバーです。

クエリの計算コストを負担するメンバーはコラボレーションごとに 1 人だけであり、そのメンバーは変更できません。

コラボレーションクリエイターが、クエリの計算コストを負担するメンバーとして誰も指定していない場合は、[クエリを行えるメンバー](#)がデフォルトの支払い担当となります。

コラボレーションで実行されたクエリの請求書は、クエリの計算コストを負担するメンバーに送られます。

クエリとジョブのコンピューティングコストを支払うメンバー

クエリとジョブのコンピューティングコストの支払いを担当するメンバー。

コラボレーションごとにクエリとジョブのコンピューティングコストを負担するメンバーは 1 人のみで、そのメンバーはイミュータブルです。

コラボレーション作成者がクエリとジョブのコンピューティングコストを支払うメンバーとして誰も指定していない場合、[クエリを行えるメンバー](#)がデフォルトの支払い者になります。

クエリとジョブのコンピューティングコストを支払うメンバーは、コラボレーションで実行されたクエリの請求書を受け取ります。

メンバーシップ

[メンバー](#)が[コラボレーション](#)に参加したときに作成されるリソースです。

メンバーがコラボレーションに関連付けるすべてのリソースは、メンバーシップの一部であるか、メンバーシップに関連付けられます。

メンバーシップを所有するメンバーだけが、そのメンバーシップのリソースを追加、削除、編集できます。

シール列

SELECT SQL 構文のために暗号化によって保護されている列です。

シードデータ

シードデータプロバイダーのデータ。[類似セグメント](#)の作成に使用されます。シードデータは直接提供することも、AWS Clean Rooms クエリの結果から取得することもできます。類似セグメントの出力は、トレーニングデータに含まれるシードユーザーに最も近いユーザーの集合です。

Spark 分析エンジン

Apache Spark SQL 関数を使用して、Amazon S3、Amazon Athena、または Snowflake に保存されている大規模なデータセットに対して複雑なクエリを AWS Clean Rooms 実行できる の分析オプション。また、 の PySpark 分析もサポートしています AWS Clean Rooms。

[CreateCollaboration API](#) を使用してコラボレーションを作成する場合、Spark 分析エンジンの値は `SPARK` です。

クエリ

サポートされている関数、クラス、変数のセットを使用して、コラボレーション内の設定済みテーブルにアクセスして分析する方法。

AWS Clean Rooms は現在、SQL クエリ言語をサポートしています。

AWS Clean Rooms は現在、直接 SQL クエリの実行または SQL 分析テンプレートを使用したクエリの実行をサポートしています。

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。