デベロッパーガイド

AWS Lake Formation

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

AWS Lake Formation: デベロッパーガイド

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 Lake Formation 1
Lake Formation の機能 2
データインジェストと管理 2
セキュリティ管理 3
Bring your data into the Data Catalog 4
使用方法 6
Lake Formation 許可管理ワークフロー 6
メタデータアクセス許可 8
ストレージアクセス管理 11
Lake Formation でのクロスアカウントデータ共有 13
Lake Formation コンポーネント 14
Lake Formation コンソール 14
Lake Formation API とコマンドラインインターフェイス 14
その他の AWS サービス 14
Lake Formation の用語 15
データレイク 15
データアクセス 15
ハイブリッドアクセスモード 15
ブループリント 15
ワークフロー 16
Data Catalog 16
基盤となるデータ 16
プリンシパル 16
データレイク管理者 17
AWS Lake Formation との サービス統合 17
追加の Lake Formation リソース 19
ブログ 19
テックトークとウェビナー 20
最新のアーキテクチャ 20
データメッシュリソース 20
ベストプラクティスガイド 20
Lake Formation の使用の開始 20
開始方法 22
初期設定 AWS タスクを完了する 22

にサインアップする AWS アカウント 22
セットアップ AWS Lake Formation 22
CloudFormation テンプレートを使用して Lake Formation リソースをセットアップする 23
データレイク管理者を作成する 24
デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する 29
Lake Formation ユーザーにアクセス許可を割り当てる 30
データレイク用の Amazon S3 ロケーションを設定する 32
(オプション) 外部データフィルタリング設定 33
(オプション) Data Catalog 暗号化キーへのアクセス権を付与する 34
(オプション) ワークフロー用の IAM ロールを作成する 34
Lake Formation モデルに対する AWS Glue データの許可のアップグレード 36
デフォルトのアクセス許可について 36
既存のアクセス許可をリストする 38
Lake Formation アクセス許可をセットアップする 40
ユーザーに IAM アクセス許可を付与する 41
Lake Formation アクセス許可モデルに切り替える 41
ステップ 5: 新しい Data Catalog リソースをセキュア化する 45
ステップ 6: ユーザーに新しい IAM ポリシーを付与する 45
ステップ 7: 既存の IAM ポリシーをクリーンアップする 47
Amazon VPC エンドポイントのセットアップ (AWS PrivateLink) 47
Lake Formation VPC エンドポイントに関する考慮事項 48
Lake Formation 用のインターフェイス VPC エンドポイントの作成 48
Lake Formation 用の VPC エンドポイントポリシーの作成 48
チュートリアル 50
AWS CloudTrail ソースからのデータレイクの作成 51
対象者 52
前提条件 53
ステップ 1: データアナリストユーザーの作成 54
ステップ 2: ワークフローロールに AWS CloudTrail ログを読み取るアクセス許可を追加する 55
ステップ 3: データレイクとしての Amazon S3 バケットを作成する 55
ステップ 4: Amazon S3 パスを登録する 56
ステップ 5: データのロケーションの許可を付与する 56
ステップ 6: Data Catalog でデータベースを作成する 57
ステップ 7: データの許可を付与する 57

ステップ 8: ブループリントを使用してワークフローを作成する 58
ステップ 9: ワークフローを実行する 59
ステップ 10: テーブルに対する SELECT を付与する 60
ステップ 11: を使用してデータレイクをクエリする Amazon Athena 61
JDBC ソースからのデータレイクの作成 62
対象者 62
前提条件 63
ステップ 1: データアナリストユーザーの作成 64
ステップ 2: AWS Glue で接続を作成する 65
ステップ 3: データレイク用の Amazon S3 バケットを作成する 65
ステップ 4: Amazon S3 パスを登録する 66
ステップ 5: データのロケーションに対する許可を付与する 66
ステップ 6: Data Catalog でデータベースを作成する 67
ステップ 7: データの許可を付与する 67
ステップ 8: ブループリントを使用してワークフローを作成する 68
ステップ 9: ワークフローを実行する 69
ステップ 10: テーブルに対する SELECT を付与する 70
ステップ 11: を使用してデータレイクをクエリする Amazon Athena 71
ステップ 12: Amazon Redshift Spectrum を使用してデータレイク内のデータをクエリする 72
ステップ 13: Amazon Redshift Spectrum を使用して Lake Formation 許可を付与または取り消す 76
Lake Formation でのオープンテーブルフォーマットのアクセス許可の設定 76
対象者 77
前提条件 78
ステップ 1: リソースをプロビジョニングする 79
ステップ 2: Iceberg テーブルのアクセス許可をセットアップする 81
ステップ 3: Hudi テーブルのアクセス許可をセットアップする 87
ステップ 4: Delta Lake テーブルのアクセス許可をセットアップする 90
ステップ 5: AWS リソースをクリーンアップする 92
タグベースのアクセスコントロールを使用したデータレイクの管理 92
対象者 94
前提条件 95
ステップ 1: リソースをプロビジョニングする 95
ステップ 2: データのロケーションを登録し、LF タグオントロジーを作成して、アクセス許可を付与する 96

ステップ 3: Lake Formation のデータベースを作成する 101
ステップ 4: テーブルの許可を付与する 111
ステップ 5: Amazon Athena でクエリを実行して許可を検証する 113
ステップ 6: AWS リソースをクリーンアップする 114
行レベルのアクセスコントロールによるデータレイクの保護 115
対象者 115
前提条件 116
ステップ 1: リソースをプロビジョニングする 117
ステップ 2: データフィルターなしでクエリを実行する 118
ステップ 3: データフィルターを設定し、許可を付与する 120
ステップ 4: データフィルターを使用してクエリを実行する 122
ステップ 5: AWS リソースをクリーンアップする 124
Lake Formation を使用してデータを安全に共有する 124
対象者 125
Lake Formation 設定を構成する 126
ステップ 1: AWS CloudFormation テンプレートを使用してリソースをプロビジョニングする 128
ステップ 2: Lake Formation クロスアカウント共有の前提条件 131
ステップ 3: タグベースのアクセスコントロール方式を使用してクロスアカウント共有を実装する 133
ステップ 4: 名前付きリソース方式を実装する 139
ステップ 5: AWS リソースをクリーンアップする 143
きめ細かなアクセスコントロール AWS アカウント を使用した外部との Data Catalog リソースの共有 144
対象者 145
前提条件 146
ステップ 1: 別のアカウントに対してきめ細かなアクセスを提供する 147
ステップ 2: 同じアカウント内のユーザーにきめ細かなアクセスを提供する 149
Lake Formation 許可へのオンボーディング 151
Lake Formation 許可の概要 152
細粒度のアクセスコントロールのための方式 154
メタデータのアクセスコントロール 157
基盤となるデータのアクセスコントロール 161
Lake Formation のペルソナと IAM 許可のリファレンス 167
AWS Lake Formation ペルソナ 167
AWS Lake Formation の マネージドポリシー 169

ペルソナに推奨される許可 177
データレイクのデフォルト設定の変更 188
黙示的な Lake Formation 許可 191
Lake Formation 許可のリファレンス 193
リソースタイプ別の Lake Formation 許可 193
Lake Formation の許可と取り消し AWS CLI コマンド 196
Lake Formation 許可 201
IAM アイデンティティセンターの統合 216
IAM アイデンティティセンターを Lake Formation と統合するための前提条件 217
Lake Formation と IAM アイデンティティセンターとの接続 220
IAM アイデンティティセンター統合の更新 225
IAM アイデンティティセンターとの Lake Formation 統合の削除 227
ユーザーおよびグループへのアクセス許可の付与 228
CloudTrail ログへの IAM アイデンティティセンターのユーザーコンテキストの追加 231
データレイクへの Amazon S3 ロケーションの追加 233
ロケーションの登録に使用されるロールの要件 234
Amazon S3 ロケーションの登録 242
暗号化された Amazon S3 ロケーションの登録 246
別の AWS アカウントでの Amazon S3 ロケーションの登録 251
AWS アカウント間で暗号化された Amazon S3 の場所を登録する 253
Amazon S3 ロケーションの登録解除 258
Amazon S3 ロケーションへのアクセス 259
ハイブリッドアクセスモード 263
一般的なハイブリッドアクセスモードのユースケース 265
ハイブリッドアクセスモードの仕組み 266
ハイブリッドアクセスモードの設定 - 一般的なシナリオ 268
ハイブリッドアクセスモードからプリンシパルとリソースを削除する 289
ハイブリッドアクセスモードでプリンシパルとリソースを表示する 290
その他のリソース 291
でのオブジェクトの作成 AWS Glue Data Catalog 291
カタログの作成 292
データベースを作成する 293
テーブルの作成 294
データカタログビューの構築 302
ワークフローを使用したデータのインポート 349
ブループリントとワークフロー 350

ワークフローの作成 351
ワークフローの実行 355
Bringing your data into the Data Catalog 357
カタログフェデレーション 359
カタログフェデレーションの機能 359
制限事項 360
へのフェデレーション Snowflake 360
へのフェデレーション Databricks 368
Bringing Amazon Redshift data into the Data Catalog 376
主な利点 379
役割と責任 379
前提条件 380
Amazon Redshift フェデレーティッドカタログの作成 386
カタログオブジェクトの表示 395
フェデレーティッドカタログの更新 397
共有フェデレーティッドカタログへのアクセス 399
フェデレーティッドカタログの削除 404
フェデレーティッドカタログのクエリ 405
その他のリソース 405
外部データソースへのフェデレーション 405
ワークフロー 406
前提条件 407
フェデレーティッドカタログの作成 412
カタログオブジェクトの表示 418
フェデレーティッドカタログの削除 420
フェデレーティッドカタログのクエリ 421
その他のリソース 421
Amazon S3 Tables と Data Catalog および Lake Formation の統合 421
データカタログと Lake Formation の統合の仕組み 423
前提条件 425
Amazon S3 Tables 統合の有効化 427
データベースとテーブルの作成 433
別の AWS アカウントに Amazon S3 テーブルバケットを登録する 436
アクセス許可の付与 439
S3 Tables 統合のアクセスコントロールの変更 442
S3 Tables 統合で Lake Formation を有効にする 443

Lake Formation から IAM への変更 ... 445
Amazon Redshift マネージドカタログの作成 ... 448
Amazon Redshift データ共有でのデータに対するアクセス許可の管理 ... 452
前提条件 ... 454
Amazon Redshift データ共有に対するアクセス許可の設定 ... 454
フェデレーションデータベースのクエリ ... 459
外部メタストアを使用するデータセットのアクセス許可の管理 ... 459
ワークフロー ... 462
前提条件 ... 463
データカタログを外部 Hive メタストアに接続する ... 465
その他のリソース ... 468
Lake Formation 許可の管理 ... 469
データロケーション許可の付与 ... 469
データロケーション許可の付与 (同じアカウント) ... 470
データロケーション許可の付与 (外部アカウント) ... 473
アカウントと共有されたデータロケーションに対する許可の付与 ... 476
データアクセス許可の付与 ... 477
Lake Formation 許可の付与に必要な IAM 許可 ... 479
名前付きリソース方式の使用 ... 482
タグベースのアクセス制御 ... 507
LF-TBAC 方式を使用したデータレイク許可の付与 ... 569
属性ベースのアクセスコントロール ... 577
許可のシナリオ例 ... 581
データフィルタリングとセルレベルのセキュリティ ... 583
データフィルター ... 585
行フィルター式での PartiQL のサポート ... 589
セルレベルのフィルタリングを使用したテーブルのクエリに必要な許可 ... 591
データフィルターの管理 ... 592
Lake Formation でのデータベースとテーブル許可の表示 ... 607
コンソールを使用した許可の取り消し ... 611
クロスアカウントデータ共有 ... 612
前提条件 ... 615
クロスアカウントデータ共有のバージョン設定の更新 ... 619
外部アカウントから AWS アカウント または IAM プリンシパル間で Data Catalog テーブルとデータベースを共有する ... 625
アカウントと共有されたデータベースまたはテーブルに対する許可の付与 ... 631

リソースリンク許可の付与 ... 633
共有テーブルの基盤となるデータへのアクセス ... 636
CloudTrail のクロスアカウントロギング ... 638
AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理 ... 643
GetResourceShares API 操作を使用したすべてのクロスアカウント付与の表示 ... 645
共有 Data Catalog テーブルとデータベースへのアクセスと表示 ... 647
AWS RAM リソース共有の招待の承諾 ... 648
共有 Data Catalog テーブルとデータベースの表示 ... 650
リソースリンクの作成 ... 652
リソースリンクの仕組み ... 653
共有テーブルへのリソースリンクの作成 ... 655
共有データベースへのリソースリンクの作成 ... 658
AWS Glue API でのリソースリンク処理 ... 662
クロスリージョンのテーブルアクセス ... 666
ワークフロー ... 668
クロスリージョンのテーブルアクセスの設定 ... 672
セキュリティ ... 675
データ保護 ... 675
保管時の暗号化 ... 676
インフラストラクチャセキュリティ ... 677
サービス間の混乱した代理の防止 ... 678
AWS Lake Formation のセキュリティイベントログ記録 ... 679
Lake Formation との統合 ... 680
Lake Formation アプリケーション統合の使用 ... 680
Lake Formation アプリケーション統合の仕組み ... 681
Lake Formation アプリケーション統合におけるロールと責任 ... 683
アプリケーション統合 API 操作の Lake Formation ワークフロー ... 684
サードパーティークエリエンジンの登録 ... 686
サードパーティのクエリエンジンがアプリケーション統合 API 操作を呼び出すアクセス許可を有効にする ... 688
フルテーブルアクセスのためのアプリケーション統合 ... 692
他の AWS サービスの使用 ... 695
Amazon Athena ... 699
トランザクションテーブル形式のサポート ... 701
その他のリソース ... 704
Amazon Redshift Spectrum ... 704

トランザクションテーブルタイプのサポート ... 705
その他のリソース ... 706
AWS Glue ... 707
トランザクションテーブルタイプのサポート ... 708
その他のリソース ... 709
Amazon EMR ... 709
トランザクションテーブル形式のサポート ... 710
その他のリソース ... 711
Quick ... 711
その他のリソース ... 712
AWS CloudTrail レイク ... 712
を使用した AWS Lake Formation API コールのログ記録 AWS CloudTrail ... 713
CloudTrail 内の Lake Formation 情報 ... 713
Lake Formation イベントについて ... 714
Lake Formation を使用した CloudTrail のデータアクセスのトレース ... 716
Lake Formation のベストプラクティス、考慮事項、制限事項 ... 719
クロスアカウントデータ共有のベストプラクティスと考慮事項 ... 719
サービスにリンクされたロールの制限 ... 722
クロスリージョンのデータアクセスに関する制限 ... 722
データカタログビューの考慮事項と制限 ... 723
データフィルタリングの制限事項 ... 724
列レベルのフィルタリングに関する注意点と制限 ... 725
セルレベルのフィルタリングの制限 ... 726
ハイブリッドアクセスモードには次の考慮事項と制限事項が適用されます。 ... 728
Amazon Redshift データウェアハウスデータを に取り込む際の制限 AWS Glue Data Catalog ... 730
S3 Tables カタログ統合の制限事項 ... 731
Hive メタデータストアのデータ共有に関する考慮事項と制限事項 ... 732
Amazon Redshift データ共有の制限事項 ... 734
IAM アイデンティティセンター 統合の制限事項 ... 735
Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項 ... 736
属性ベースのアクセス制御に関する考慮事項、制限、サポートされているリージョン ... 739
Lake Formation のトラブルシューティング ... 741
一般的なトラブルシューティング ... 741
エラー: Insufficient Lake Formation permissions on <Amazon S3 location> (<Amazon S3 のロケーション> に対する Lake Formation 許可が不十分です) ... 741

エラー:「Insufficient encryption key permissions for Glue API」(Glue API の暗号化キー許可が不十分です) 742
マニフェストを使用する自分のクエリ Amazon Athena または Amazon Redshift クエリが失敗している 742
エラー:「Insufficient Lake Formation permission(s): Required create tag on catalog」(Lake Formation 許可が不十分です: カタログに対する必須の create タグ) 742
無効なデータレイク管理者を削除するとエラーが発生します 742
エラー: "Caller is temporarily not authorized to perform this operation on resource..." 742
クロスアカウントアクセスのトラブルシューティング 742
クロスアカウント Lake Formation 許可を付与しましたが、受領者がリソースを表示できません 743
受領者アカウントのプリンシパルは、Data Catalog リソースを表示することはできますが、基盤となるデータにはアクセスできません。 744
エラー: AWS RAM リソース共有の招待を受け入れるときに、「発信者が承認されなかったため関連付けに失敗しました」 744
エラー:「Not authorized to grant permissions for the resource」(リソースの許可を付与する権限がありません) 745
エラー: AWS 「組織情報を取得するためのアクセスが拒否されました」 745
エラー:「Organization <organization-ID> not found」(組織 <organization-ID> が見つかりません) 745
エラー:「Insufficient Lake Formation permissions: Illegal combination」(Lake Formation 許可が不十分です: 不正な組み合わせ) 745
外部アカウントへのリクエストを許可/取り消ししたときに発生する ConcurrentModificationException 746
Amazon EMR を使用して、クロスアカウント経由で共有されたデータにアクセスする際のエラー 746
ブループリントとワークフローのトラブルシューティング 747
ブループリントが「User: <user-ARN> is not authorized to perform: iam:PassRole on resource: <role-ARN>」(ユーザー: <user-ARN> にはリソース: <role-ARN> で iam:PassRole を実行する許可がありません) エラーで失敗しました 748
ワークフローが「User: <user-ARN> is not authorized to perform: iam:PassRole on resource: <role-ARN>」(ユーザー: <user-ARN> にはリソース: <role-ARN> で iam:PassRole を実行する許可がありません) エラーで失敗しました 748
ワークフローのクローラが「Resource does not exist or requester is not authorized to access requested permissions」(リソースが存在しないかリクエストされた認可にアクセスする権限がリクエスト元にありません) エラーで失敗しました 748

ワークフローのクローラが「An error occurred (AccessDeniedException) when calling the CreateTable operation...」(CreateTable 操作の呼び出し時にエラーが発生しました (AccessDeniedException)) で失敗しました ... 749
の既知の問題 AWS Lake Formation ... 749
テーブルメタデータのフィルタリングの制限 ... 749
除外された列の名前変更に関する問題 ... 750
CSV テーブルの列の削除に関する問題 ... 750
テーブルパーティションを共通パスの下に追加する必要性 ... 751
ワークフロー作成時におけるデータベースの作成に関する問題 ... 751
ユーザーの削除後での再作成に関する問題 ... 751
Data Catalog API 操作が IsRegisteredWithLakeFormation パラメータの値を更新しない ... 752
Lake Formation オペレーションは AWS Glue スキーマレジストリをサポートしていません ... 752
エラーメッセージを更新しました ... 752
Lake Formation API ... 753
アクセス許可 ... 754
— 操作 — ... 754
— データ型 — ... 754
データレイク設定 ... 755
— 操作 — ... 755
— データ型 — ... 755
IAM アイデンティティセンターの統合 ... 755
— 操作 — ... 755
— データ型 — ... 755
ハイブリッドアクセスモード ... 756
— 操作 — ... 756
— データ型 — ... 754
認証情報供給 ... 756
— 操作 — ... 756
— データ型 — ... 757
タグ付け ... 757
— 操作 — ... 757
— データ型 — ... 758
データフィルター API ... 758
— 操作 — ... 758

— データ型 — 758
一般的なデータ型 759
ErrorDetail 759
文字列パターン 759
サポートされるリージョン 760
一般提供 760
AWS GovCloud (US) 760
トランザクションとストレージの最適化 760
ドキュメント履歴 763
AWS 用語集 778
..... dcclxxix

とは AWS Lake Formation

AWS Lake Formation デベロッパーガイドへようこそ。

AWS Lake Formation は、分析と機械学習のためにデータを一元管理、保護、グローバルに共有できるようにします。Lake Formation では、Amazon Simple Storage Service (Amazon S3) 上のデータレイクデータと AWS Glue Data Catalogの関連メタデータに対するきめ細かなアクセスコントロールを管理できます。

Lake Formation は、IAM 許可モデルを補強する独自の許可モデルを提供します。Lake Formation アクセス許可モデルを使用すると、リレーショナルデータベース管理システム (RDBMS) と同様に、シンプルな許可または取り消しメカニズムを通じて、データレイクに保存されているデータだけでなく、Amazon Redshift データウェアハウス、 Amazon DynamoDB データベース、サードパーティーデータソースなどの外部データソースへのきめ細かなアクセスが可能になります。Lake Formation のアクセス許可は、Amazon Athena、、Amazon Redshift Spectrum、Amazon EMR、 などの AWS 分析および機械学習サービス全体で、列 Amazon Quick、行、セルレベルで詳細な制御を使用して適用されます AWS Glue。

AWS Glue Data Catalog (データカタログ) の Lake Formation ハイブリッドアクセスモードでは、Amazon S3 および アクションの Lake Formation アクセス許可と IAM アクセス許可ポリシーの両方を使用して、カタログ化されたデータを保護して AWS Glue アクセスできます。ハイブリッドアクセスモードを使用すると、データ管理者は一度に 1 つのデータレイクのユースケースに絞って、選択的かつ段階的に Lake Formation のアクセス許可をオンボーディングできます。

Lake Formation では、複数の AWS 、組織全体、または別のアカウントの IAM プリンシパルとデータを内部および外部で共有し AWS アカウント、Data Catalog メタデータと基盤となるデータにきめ細かなアクセスを提供することもできます。

トピック

- Lake Formation の機能
- AWS Lake Formation: 仕組み
- Lake Formation コンポーネント
- Lake Formation の用語
- AWS Lake Formation との サービス統合
- 追加の Lake Formation リソース
- Lake Formation の使用の開始

Lake Formation の機能

Lake Formation は、データサイロを分解し、異なるタイプの構造化および非構造化データを一元化されたリポジトリに統合するために役立ちます。まず、Amazon S3、またはリレーショナルおよび NoSQL データベース内の既存のデータストアを特定し、データをデータレイクに移動させます。その後、分析のためにデータのクロール、カタログ化、および準備を行います。次に、ユーザーが選択した分析サービス経由でのデータへのセキュアなセルフサービスアクセスをユーザーに提供します。

Lake Formation コンソールを使用して、データカタログにマルチレベルフェデレーティッドカタログを作成し、Amazon S3 データレイクと Amazon Redshift データウェアハウス間でデータを統合できます。また、 などの運用データベースや Amazon DynamoDB、Google BigQuery、MySQL などのサードパーティーデータソースからのデータを統合することもできます。データカタログは、一元化されたメタデータリポジトリを提供し、異種システム間でのデータの管理と発見を容易にします。

詳細については、「へのデータの取り込み AWS Glue Data Catalog」を参照してください。

トピック

- データインジェストと管理
- セキュリティ管理
- Bring your data into the Data Catalog

データインジェストと管理

既に にあるデータベースからデータをインポートする AWS

既存のデータベースの場所を指定し、アクセス認証情報を指定すると、Lake Formation がデータソースの内容を理解するためにデータとそのメタデータ (スキーマ) を読み取ります。その後、Lake Formation がデータを新しいデータレイクにインポートし、メタデータを中央カタログに記録します。Lake Formation を使用することで、Amazon RDS で実行されている、または Amazon EC2 でホストされている MySQL、PostgreSQL、SQL Server、MariaDB、および Oracle データベースからデータをインポートできます。データのロードは一括と増分の両方がサポートされています。

その他の外部ソースからデータをインポートする

Lake Formation は、Java Database Connectivity (JDBC) を使用した接続によるオンプレミスデータベースからのデータの移動に使用できます。コンソールでターゲットソースを特定し、アクセス認証情報を提供すると、Lake Formation がデータを読み取って、データレイクにロードします。上記の

データベース以外のデータベースからデータをインポートするには、 を使用してカスタム ETL ジョブを作成できます AWS Glue。

データをカタログ化してラベル付けする

AWS Glue クローラを使用して Amazon S3 でデータを読み取ってデータベースとテーブルスキーマを抽出し、そのデータを検索可能なデータカタログに保存できます。次に、Lake Formation Lake Formation のタグベースのアクセス制御 (TBAC) を使用して、データベース、テーブル、列に対するアクセス許可を管理します。Data Catalog へのテーブルの追加に関する詳細については、「でのオブジェクトの作成 AWS Glue Data Catalog」を参照してください。

セキュリティ管理

アクセスコントロールを定義して管理する

Lake Formation では、データレイク内のデータに対するアクセスコントロールを 1 か所で管理できます。データベース、テーブル、列、行、およびセルレベルでデータへのアクセスを制限するセキュリティポリシーを定義できます。これらのポリシーは、IAM ユーザーとロール、および外部のアイデンティプロバイダー経由でフェデレーションするユーザーとグループに適用されます。きめ細かなコントロールを使用して、Amazon Redshift Spectrum、Athena、 AWS Glue ETL、Amazon EMR for Apache Spark 内の Lake Formation で保護されたデータにアクセスできます。IAM ID を作成するときは常に、IAM ベストプラクティスに従うようにしてください。詳細については、「IAM ユーザーガイド」の「セキュリティベストプラクティス」を参照してください。

ハイブリッドアクセスモード

Lake Formation ハイブリッドアクセスモードは、データカタログ内のデータベースとテーブルに対して Lake Formation アクセス許可を選択的に有効にする柔軟性を提供します。ハイブリッドアクセスモードを使用すると、他の既存のユーザーやワークロードのアクセス許可ポリシーを中断することなく、特定のユーザーのセットに Lake Formation 許可を設定できる増分パスが導入されました。詳細については、「ハイブリッドアクセスモード」を参照してください。

監査ロギングを実装する

Lake Formation は、アクセスを監視し、一元的に定義されたポリシーへのコンプライアンスを証明するために、CloudTrail を使用した包括的な監査ログを提供します。Lake Formation を介してデータレイク内のデータを読み取る分析および機械学習サービス全体のデータアクセス履歴を監査できます。この機能により、どのユーザーまたはロールが、どのサービスを使用して、どのデータにいつアクセスしようとしたのかを確認することができます。監査ログには、CloudTrail API とコンソールを使用して他の CloudTrail ログにアクセスするのと同じ方法でアクセスできます。CloudTrail ログの詳

細については、「を使用した AWS Lake Formation API コールのログ記録 AWS CloudTrail」を参照してください。

行およびセルレベルのセキュリティ

Lake Formation は、列と行の組み合わせに対するアクセスの制限を可能にするデータフィルターを提供します。行およびセルレベルのセキュリティを使用して、個人を特定できる情報 (PII) などの機密データを保護します。行レベルのセキュリティに関する詳細については、「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」を参照してください。

タグベースのアクセス制御

Lake Formation 属性ベースのアクセスコントロールを使用して、LF タグと呼ばれるカスタムラベルを作成して、数百または数千のデータアクセス許可を管理します。LF タグを定義して、データベース、テーブル、または列にアタッチできるようになりました。次に、分析、機械学習 (ML)、および抽出、変換、ロード (ETL) サービス間で制御されたアクセスを共有して利用します。LF タグを使用すると、何千ものリソースのポリシー定義をいくつかの論理タグに置き換えることで、データガバナンスを簡単にスケールできます。Lake Formation は、このメタデータに対するテキストベースの検索機能を提供するため、ユーザーは分析する必要があるデータをすばやく見つけることができます。

属性ベースのアクセスコントロール

属性ベースのアクセスコントロールを使用して、Data Catalog オブジェクトへのアクセスを許可します。属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。 はこれらの属性タグを AWS 呼び出します。ABAC を使用して、同じアカウント内、またはデータカタログリソース上の別のアカウント内のプリンシパルにアクセスを許可できます。一致する IAM タグまたはセッションタグのキーと値を持つすべての IAM プリンシパルがリソースにアクセスできます。これらの許可を行うには、リソースに対する付与可能なアクセス許可が必要です。

クロスアカウントアクセス

Lake Formation のアクセス許可管理機能は、一元化されたアプローチを通じて複数の AWS アカウントにわたる分散データレイクの保護と管理を簡素化し、データカタログと Amazon S3 ロケーションへのきめ細かなアクセスコントロールを提供します。詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

Bring your data into the Data Catalog

フェデレーション機能を使用すると、データやメタデータを Amazon S3 または に移行することなく、フェデレーションカタログを作成し、Amazon Redshift などのさまざまなデータソースに保存さ

れているデータセットに対するアクセス許可を設定できます AWS Glue Data Catalog。次の方法を使用して、Lake Formation で外部データセットに対するデータの取得とアクセス許可の管理を行うことができます。

詳細については、「AWS Glue Data Catalogへのデータの取り込み」を参照してください。

- Amazon Redshift データウェアハウスのデータの への取り込み AWS Glue Data Catalog – 既存の Amazon Redshift 名前空間またはクラスターをデータカタログに登録し、データカタログにマルチレベルフェデレーティッドカタログを作成します。

  Amazon EMR Serverless や Amazon Athena などの、Apache Iceberg REST カタログ OpenAPI 仕様と互換性のある任意のクエリエンジンを使用してデータにアクセスできます。

  詳細については、「Amazon Redshift データを に取り込む AWS Glue Data Catalog」を参照してください。

- 外部データソースから Data Catalog にフェデレーションする – AWS Glue 接続を使用して Data Catalog を外部データソースに接続し、フェデレーションカタログを作成して Lake Formation を使用してデータセットへのアクセス許可を一元管理します。データカタログへのメタデータの移行は不要です。

  詳細については、「の外部データソースへのフェデレーション AWS Glue Data Catalog」を参照してください。

- Amazon S3 テーブルバケットとデータカタログの統合 – Amazon S3 テーブルをデータカタログオブジェクトとして公開およびカタログ化し、Lake Formation コンソールまたは AWS Glue APIs を使用して、カタログを Lake Formation データの場所として登録できます。

  詳細については、「Amazon S3 Tables と AWS Glue Data Catalog および の統合 AWS Lake Formation」を参照してください。

- データカタログで Amazon Redshift テーブルを管理するカタログを作成する – 現在、Amazon Redshift プロデューサークラスターや Amazon Redshift データ共有を使用できない場合がありますが、Data Catalog を使用して Amazon Redshift テーブルを作成および管理する必要があります。`glue:CreateCatalog` API または AWS Lake Formation コンソールを使用して AWS Glue マネージドカタログを作成するには、カタログタイプを `Catalog source` Redshift として設定`Managed`します。

  詳細については、「での Amazon Redshift マネージドカタログの作成 AWS Glue Data Catalog」を参照してください。

- Lake Formation と Amazon Redshift データ共有の統合 – Lake Formation を使用すると、Amazon Redshift データ共有のデータベース、テーブル、列、および行レベルのアクセス許可を一元管理し、データ共有内のオブジェクトへのユーザーアクセスを制限できます。
- Data Catalog を外部メタストアに接続する – 外部メタストア AWS Glue Data Catalog に接続して、Lake Formation を使用して Amazon S3 のデータセットに対するアクセス許可を管理します。データカタログへのメタデータの移行は不要です。

  詳細については、「外部メタストアを使用するデータセットのアクセス許可の管理」を参照してください。
- Lake Formation と AWS Data Exchange の統合 – Lake Formation は、 を介したデータへのアクセスのライセンスをサポートしています AWS Data Exchange。Lake Formation データのライセンスに関心をお持ちの場合は、AWS Data Exchange ユーザーガイドの「AWS Data Exchangeとは」を参照してください。

AWS Lake Formation: 仕組み

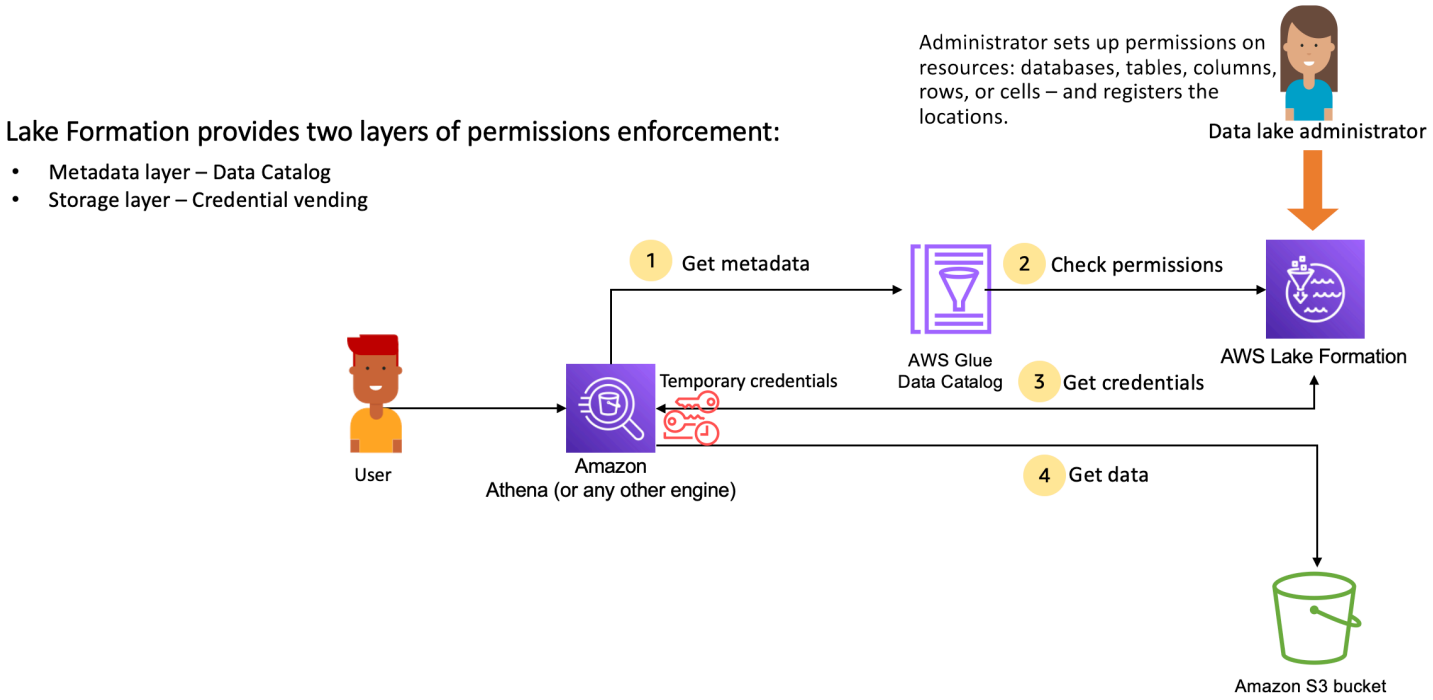
AWS Lake Formation は、Amazon S3 の基盤となるデータを持つデータベース、テーブル、列などの Data Catalog リソースへのアクセスを許可または取り消すためのリレーショナルデータベース管理システム (RDBMS) アクセス許可モデルを提供します。管理が簡単な Lake Formation 許可は、複雑な Amazon S3 バケットポリシーや対応する IAM ポリシーに取って代わるものです。

Lake Formation では、次の 2 つのレベルでアクセス許可を実装できます。

- データベースやテーブルなどのデータカタログリソースに対するメタデータレベルでアクセス許可を適用
- 統合されたエンジンに代わって、Amazon S3 に保存されている基盤となるデータへのアクセス許可を管理

Lake Formation 許可管理ワークフロー

Lake Formation は、Lake Formation に登録されている Amazon S3 データストアやメタデータオブジェクトに対してクエリを実行するために、分析エンジンと統合します。以下の図は、Lake Formation における許可管理の仕組みを示しています。



Lake Formation 許可管理の手順の概要

Lake Formation がデータレイク内のデータに対するアクセス制御を提供する前に、データレイク管理者または管理権限を持つユーザーが、Lake Formation の権限を使用して Data Catalog テーブルへのアクセスを許可または拒否する個々の Data Catalog テーブルのユーザーポリシーを設定します。

次に、データレイク管理者または管理者から委任されたユーザーのいずれかが、Data Catalog データベースとテーブルに対するユーザーに Lake Formation 許可を付与し、テーブルの Amazon S3 ロケーションを Lake Formation に登録します。

1. メタデータの取得 – プリンシパル (ユーザー) は、Amazon Athena、Amazon EMR AWS Glue、Amazon Redshift Spectrum などの統合分析エンジンにクエリまたは ETL スクリプトを送信します。統合分析エンジンは、要求されているテーブルを識別し、メタデータのリクエストを Data Catalog に送信します。
2. 許可の確認 — Data Catalog は Lake Formation でユーザーのアクセス許可を確認し、ユーザーがテーブルにアクセスする権限を持っている場合は、ユーザーが表示できるメタデータをエンジンに返します。
3. 認証情報の取得 — Data Catalog は、テーブルが Lake Formation によって管理されているかどうかをエンジンに知らせます。基盤となるデータが Lake Formation に登録されている場合、分析エンジンは Lake Formation に一時的なアクセスを許可してデータアクセスを提供するように要求します。

4. データの取得 — ユーザーがテーブルへのアクセスを許可されている場合、Lake Formation は統合分析エンジンへの一時的なアクセスを提供します。一時的なアクセスを使用して、分析エンジンは Amazon S3 からデータを取得し、列、行、またはセルのフィルタリングなど、必要なフィルタリングを実行します。エンジンはジョブの実行を終了すると、結果をユーザーに返します。このプロセスは、認証情報の供給と呼ばれます。

   テーブルが Lake Formation によって管理されていない場合、分析エンジンからの 2 回目の呼び出しは Amazon S3 に対して直接行われます。関係する Amazon S3 バケットポリシーと IAM ユーザーポリシーのデータアクセスが評価されます。

   IAM ポリシーを使用するときは、常に IAM のベストプラクティスに従うようにしてください。詳細については、「IAM ユーザーガイド」の「IAM でのセキュリティベストプラクティス」を参照してください。

トピック

- メタデータアクセス許可
- ストレージアクセス管理
- Lake Formation でのクロスアカウントデータ共有

メタデータアクセス許可

Lake Formation は、Data Catalog の認可とアクセスコントロールを行います。IAM ロールが任意のシステムから Data Catalog API 呼び出しを行うと、Data Catalog はユーザーのデータの許可を検証し、ユーザーがアクセス許可を持っているメタデータのみを返します。例えば、IAM ロールがデータベース内の 1 つのテーブルにのみアクセスでき、そのロールを引き受けるサービスまたはユーザーが `GetTables` 操作を実行した場合、データベース内のテーブルの数に関係なく、レスポンスには 1 つのテーブルのみが含まれます。

デフォルト設定 - **`IAMAllowedPrincipal`** グループのアクセス許可

AWS Lake Formationは、デフォルトでは、すべてのデータベースとテーブルのアクセス許可を `IAMAllowedPrincipal` という名前の仮想グループに設定します。このグループは一意で、Lake Formation 内でのみ見ることができます。`IAMAllowedPrincipal` グループには、IAM プリンシパルポリシーとリソースポリシーを通じて Data Catalog リソースにアクセスできるすべての IAM プリンシパルが含まれます AWS Glue 。このアクセス許可がデータベースまたはテーブルに存在する場合、すべてのプリンシパルにデータベースまたはテーブルへのアクセス許可が付与されます。

データベースまたはテーブルに対してより詳細なアクセス許可を与える場合は、IAMAllowedPrincipal 許可を削除すると、Lake Formation はそのデータベースまたはテーブルに関連する他のすべてのポリシーを適用します。例えば、ユーザー A が DESCRIBE 許可でデータベース A にアクセスすることを許可するポリシーがあり、IAMAllowedPrincipal がすべての許可で存在する場合、ユーザー A は IAMAllowedPrincipal 許可が取り消されるまで、他のすべてのアクションを実行し続けます。

さらに、デフォルトでは、IAMAllowedPrincipal グループは、新しいデータベースとテーブルの作成時に、すべての許可を持っています。この動作を制御する設定は 2 つあります。1 つ目はアカウントとリージョンレベルで、新しく作成されたデータベースに対してこれを有効にするもので、2 つ目はデータベースレベルです。デフォルト設定を変更するには、「デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する」を参照してください。

アクセス許可の付与

データレイク管理者は、プリンシパルに Data Catalog 許可を付与して、プリンシパルがデータベースとテーブルを作成および管理し、基盤となるデータにアクセスできるようにすることができます。

データベースとテーブルレベルのアクセス許可

Lake Formation 内で許可を付与する場合、付与者はアクセス許可を付与するプリンシパル、アクセス許可を付与するリソース、および付与対象者が実行できるアクセス権を持つべきアクションを指定する必要があります。Lake Formation 内のほとんどのリソースについて、権限を付与するプリンシパルリストとリソースは同様ですが、被付与者が実行できるアクションはリソースタイプによって異なります。例えば、テーブルに対しては、テーブルを読み取るための SELECT 許可が利用できますが、データベースに対しては SELECT 許可は利用できません。CREATE_TABLE 許可は、データベースではアクセス許可することができますが、テーブルではアクセス許可できません。

次の 2 つの方法を使用して AWS Lake Formation アクセス許可を付与できます。

- 名前付きリソースメソッド — ユーザーに許可を付与する際に、データベースとテーブルの名前を選択できます。
- LF タグベースのアクセス制御 (LF-TBAC) — ユーザーは LF タグを作成し、それらを Data Catalog リソースに関連付け、LF タグに対する Describe 許可を付与し、個々のユーザーにアクセス許可を関連付け、LF タグを使用して LF 許可ポリシーをさまざまなユーザーに書き込みます。このような LF タグベースのポリシーは、それらの LF タグ値に関連付けられているすべての Data Catalog リソースに適用されます。

> **Note**
>
> LF タグは Lake Formation に固有のものです。これらは Lake Formation でのみ表示され、AWS リソースタグと混同しないでください。

LF-TBAC は、ユーザーがリソースをユーザー定義の LF タグカテゴリにグループ化し、それらのリソースグループにアクセス許可を適用できるようにする機能です。したがって、これは膨大な数の Data Catalog リソースにわたってアクセス許可をスケーリングする最適な方法です。

詳細については、「Lake Formation のタグベースのアクセス制御」を参照してください。

プリンシパルにアクセス許可を付与すると、Lake Formation はアクセス許可をそのユーザーのすべてのポリシーの統合として評価します。例えばテーブルにプリンシパル用の 2 つのポリシーがあり、一方のポリシーが名前付きリソースメソッドを使用して列 col1、col2、col3 に許可を付与し、もう一方のポリシーが LF タグを使用して同じテーブルとプリンシパルへのアクセス許可を col5 と col6 に付与する場合、有効なアクセス許可は col1、col2、col3、col5、col6 という許可の和になります。これにはデータフィルターと行も含まれます。

データロケーション許可

データロケーション許可は、管理者以外のユーザーが特定の Amazon S3 の ロケーションに対してデータベースとテーブルを作成できるようにします。作成するためのアクセス許可のない場所にユーザーがデータベースまたはテーブルを作成しようとすると、作成タスクは失敗します。これは、ユーザーがデータレイク内の任意の場所にテーブルを作成することを防ぎ、ユーザーがデータを読み書きできる場所を制御できるようにするためです。作成先のデータベース内の Amazon S3 ロケーションにテーブルを作成する場合、暗黙的なアクセス許可が存在します。詳細については、「データロケーション許可の付与」を参照してください。

テーブルとデータベースのアクセス許可の作成

管理者以外のユーザーには、デフォルトではデータベースまたはデータベース内のテーブルを作成する権限がありません。データベースの作成は、権限のあるプリンシパルのみがデータベースを作成できるように、Lake Formation 設定を使用してアカウントレベルで制御されます。詳細については、「データベースを作成する」を参照してください。テーブルを作成するには、プリンシパルにはテーブルが作成されているデータベースに対する CREATE_TABLE 許可が必要です。詳細については、「テーブルの作成」を参照してください。

暗黙的なアクセス許可および明示的なアクセス許可

Lake Formation では、ペルソナとペルソナが実行するアクションに応じて暗黙的なアクセス許可が提供されます。例えば、データレイク管理者は、Data Catalog 内のすべてのリソースに対する DESCRIBE 許可、すべての口ケーションに対するデータロケーション許可、すべての口ケーションのデータベースとテーブルの作成許可、および任意のリソースに対する Grant 許可と Revoke 許可を自動的に取得します。データベース作成者は作成したデータベースに対するすべてのデータベース許可を自動的に取得し、テーブル作成者は作成したテーブルに対するすべてのアクセス許可を取得します。詳細については、「黙示的な Lake Formation 許可」を参照してください。

付与可能なアクセス許可

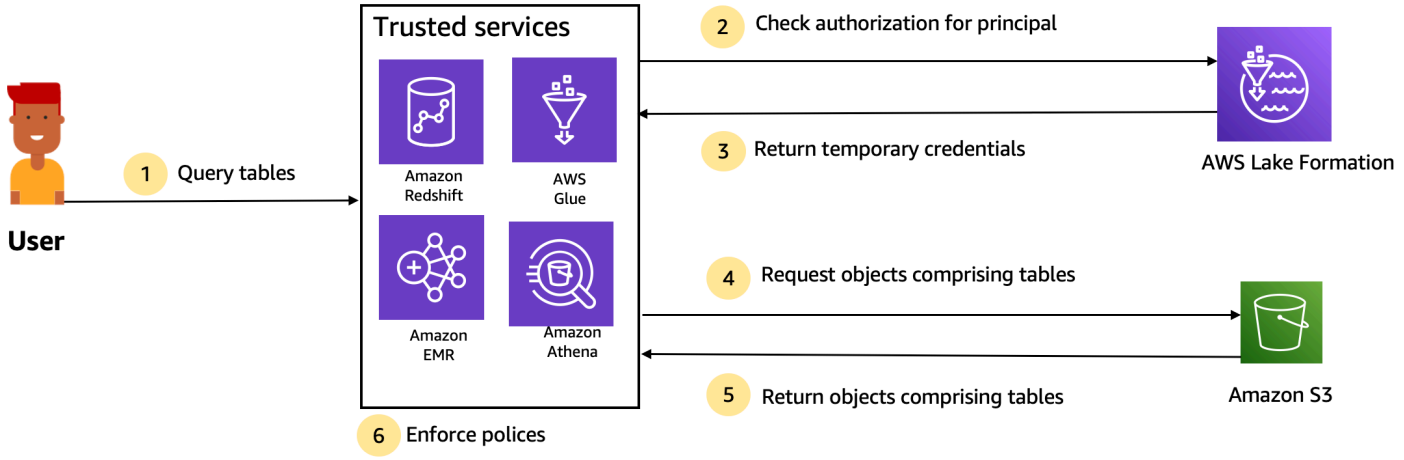
データレイク管理者は、付与可能なアクセス許可を付与することで、管理者以外のユーザーにアクセス許可の管理を委任することができます。プリンシパルにリソースに対する付与可能なアクセス許可と一連のアクセス許可が与えられると、そのプリンシパルはそのリソースの他のプリンシパルにアクセス許可を付与できるようになります。

ストレージアクセス管理

Lake Formation では、認証情報の供給を使用して Amazon S3 データへの一時的なアクセスを提供します。認証情報の供給、またはトークンの供給は、リソースへの短期アクセスを許可する目的で、ユーザー、サービス、またはその他のエンティティに一時的な認証情報を提供する一般的なパターンです。

Lake Formation はこのパターンを活用して、呼び出し元のプリンシパルに代わってデータにアクセスするための Athena などの AWS 分析サービスへの短期アクセスを提供します。アクセス許可を付与する際、ユーザーは Amazon S3 バケットポリシーや IAM ポリシーを更新する必要はなく、Amazon S3 に直接アクセスする必要もありません。

次の図は、Lake Formation が登録された場所への一時的なアクセスを提供する方法を示しています。



Trusted services enforce AWS Lake Formation policies (distributed enforcement with fail close).

1. プリンシパル (ユーザー) は、Athena、Amazon EMR、Redshift Spectrum、 AWS Glueなどの信頼できる統合サービスを通じて、テーブルのデータを求めるクエリまたはリクエストを入力します。
2. 統合サービスは、テーブルと要求された列についてLake Formation からの承認を確認し、承認の決定を行います。ユーザーに権限がない場合、Lake Formation はデータへのアクセスを拒否し、クエリは失敗します。
3. 認可が成功し、テーブルとユーザーのストレージ認可が有効になると、統合サービスは Lake Formation から一時的な認証情報を取得してデータにアクセスします。
4. 統合サービスは、Lake Formation の一時的な認証情報を使用して Amazon S3 にオブジェクトを要求します。
5. Amazon S3 は、統合されたサービスに Amazon S3 オブジェクトを提供します。Amazon S3 オブジェクトには、テーブルのすべてのデータが含まれています。
6. 統合サービスは、列レベル、行レベル、セルレベルのフィルタリングなど、必要な Lake Formation ポリシーの適用を行います。統合サービスがクエリを処理し、ユーザーに結果を返します。

Data Catalog テーブルに対してストレージレベルの許可の適用を有効にする

デフォルトでは、Data Catalog 内のテーブルではストレージレベルの適用は有効になっていません。ストレージレベルの適用を有効にするには、ソースデータの Amazon S3 ロケーションを Lake Formation に登録し、IAM ロールを提供する必要があります。ストレージレベルのアクセス許可は、Amazon S3 ロケーションの同じテーブルロケーションのパスまたはプレフィックスを持つすべてのテーブルに対して有効になります。

統合サービスがユーザーに代わってデータロケーションへのアクセスを要求すると、Lake Formation サービスがこの役割を引き受け、要求されたサービスにリソースへのスコープダウンされたアクセス許可を持つ認証情報を返し、データアクセスができるようにします。登録された IAM ロールには、AWS KMS キーを含む Amazon S3 ロケーションへのすべての必要なアクセス権が必要です。

詳細については、「Amazon S3 ロケーションの登録」を参照してください。

サポートされている AWS サービス

AWS Athena、Redshift Spectrum、Amazon EMR、 などの分析サービス AWS Glue Amazon Quick。 AWS Lake Formation 認証情報供給 API オペレーションを使用して Lake Formation と Amazon SageMaker AI 統合します。Lake Formation と統合する AWS サービスの完全なリスト、およびそれらがサポートする詳細度とテーブル形式については、「」を参照してください他の AWS サービスの使用。

Lake Formation でのクロスアカウントデータ共有

Lake Formation では、名前付きリソース方式や LF タグを使った簡単な設定で、 AWS アカウント内やアカウント間で Data Catalog リソース (データベースやテーブル) を共有することができます。データベース全体を共有したり、データベースからテーブルをアカウント内の任意の IAM プリンシパル (IAM ロールとユーザー) 、アカウントレベルの他の AWS アカウント、または別のアカウントの IAM プリンシパルに直接選択したりできます。

Data Catalog テーブルをデータフィルターと共有して、行レベルとセルレベルの詳細へのアクセスを制限することもできます。Lake Formation は AWS Resource Access Manager (AWS RAM) を使用して、アカウント間のアクセス許可の付与を容易にします。リソースを 2 つのアカウントで共有すると、 AWS RAM は受信者アカウントに招待状を送信します。ユーザーが AWS RAM 共有の招待を受け入れると、 は、データカタログリソースを使用可能にし、ストレージレベルの強制を有効にするために必要なアクセス許可を Lake Formation AWS RAM に提供します。詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

受信者アカウントのデータレイク管理者が AWS RAM 共有を受け入れると、共有リソースは受信者アカウントで使用できます。データレイク管理者は、管理者が共有リソースに対して GRANTABLE 許可を持っている場合、受信者アカウントの追加の IAM プリンシパルに、共有リソースに対してさらに Lake Formation 許可を付与します。

ただし、プリンシパルは、リソースリンクがないと Athena または Redshift Spectrum を使用して共有リソースをクエリすることはできません。リソースリンクは Data Catalog 内のエンティティであり、Linux-Symlink の概念に似ています。

受信者アカウントのデータレイク管理者が、共有リソースにリソースリンクを作成します。管理者は、元の共有リソースに必要な許可とともに、リソースリンクの `Describe` 許可を追加のユーザーに付与します。受信者アカウントのユーザーは、リソースリンクを使用して Athena と Redshift Spectrum を使用して共有リソースをクエリできます。リソースリンクの詳細については、「リソースリンクの作成」を参照してください。

Lake Formation コンポーネント

AWS Lake Formation は、複数のコンポーネントとの相互作用によって、データレイクの作成と管理を行っています。

Lake Formation コンソール

Lake Formation コンソールを使用して、データレイクの定義と管理、および Lake Formation 許可の付与と取り消しを行います。コンソールでブループリントを使用して、データの検出、クレンジング、変換、および取り込みを行うことができます。個々の Lake Formation ユーザーに対してコンソールへのアクセスを有効化または無効化することもできます。

Lake Formation API とコマンドラインインターフェイス

Lake Formation は、複数の言語固有の SDK と AWS Command Line Interface (AWS CLI) を使用して API 操作を提供します。Lake Formation API は AWS Glue API と連携して動作します。Lake Formation API は主に Lake Formation 許可の管理に焦点を当てる一方で、AWS Glue APIはデータカタログ APIと、データに対する ETL 操作の定義、スケジュール、および実行のためのマネージドインフラストラクチャを提供します。

AWS Glue APIについては、「AWS Glue デベロッパーガイド」を参照してください。AWS CLI の使用については、「AWS CLI コマンドリファレンス」を参照してください。

その他の AWS サービス

Lake Formation は、以下のサービスを利用します。

- AWS Glue 変換を使用してデータを変換するジョブとクローラをオーケストレートするための AWS Glue
- Lake Formation プリンシパルに許可ポリシーを付与するための IAM Lake Formation の許可モデルは、データレイクをセキュア化するために IAM 許可モデルを補強します。

Lake Formation の用語

以下は、本ガイドで使用される重要な用語の一部です。

データレイク

データレイクは、Amazon S3 に保存され、データカタログを使用して Lake Formation によって管理される永続的なデータです。通常、データレイクには以下のデータが保存されます。

- 構造化データと非構造化データ
- raw データと変換されたデータ

Amazon S3 パスをデータレイク内に配置するには、パスを Lake Formation に登録する必要があります。

データアクセス

Lake Formation は、AWS Identity and Access Management (IAM) ポリシーを補強する新しい許可/取り消しモデルを通じて、データへのセキュアで詳細なアクセスを提供します。

アナリストとデータサイエンティストは、Amazon Athena などの AWS の分析および機械学習サービスの完全なポートフォリオを使用して、データにアクセスすることができます。設定済みの Lake Formation のセキュリティポリシーは、ユーザーがアクセスを認可されているデータにしかアクセスできないことを確実にするために役立ちます。

ハイブリッドアクセスモード

ハイブリッドアクセスモードでは、Lake Formation のアクセス許可と IAM や Amazon S3 のアクセス許可の両方を使用して、カタログ化されたデータを保護およびアクセスできます。ハイブリッドアクセスモードを使用すると、データ管理者は、一度に 1 つのデータレイクのユースケースに絞って、選択的かつ段階的に Lake Formation のアクセス許可をオンボーディングできます。

ブループリント

ブループリントとは、データレイクに簡単にデータを取り込むことができるデータ管理テンプレートです。Lake Formation は、リレーショナルデータベースや AWS CloudTrail ログなどの事前定義されたソースタイプごとに、いくつかのブループリントを提供します。ブループリントからは、ワークフローを作成できます。ワークフローは、データのロードと更新をオーケストレートするために生成される、AWS Glue のクローラー、ジョブ、およびトリガーで構成されます。ブループリントは、デー

タソース、データターゲット、およびスケジュールを入力として使用して、ワークフローを設定します。

ワークフロー

ワークフローは、一連の関連する AWS Glue ジョブ、クローラ、トリガーをまとめるコンテナです。Lake Formation でワークフローを作成すると、それが AWS Glue サービスで実行されます。Lake Formation は、ワークフローのステータスを単一のエンティティとして追跡できます。

ワークフローを定義するときは、ワークフローの基礎となるブループリントを選択します。その後、ワークフローをオンデマンドで、またはスケジュールに従って実行できます。

Lake Formation で作成するワークフローは、AWS Glue コンソールに DAG (Directed Acyclic Graph) として表示されます。DAG を使用することで、ワークフローの進行状況を追跡し、トラブルシューティングを実行できます。

Data Catalog

データカタログは、永続的なメタデータストアです。これは、AWS クラウド上でのメタデータの保存、注釈付け、および共有を、Apache Hive メタストアと同じ方法で実行できるマネージドサービスです。異種システムがデータサイロ内のデータを追跡するためのメタデータを保存して検索できる均一なリポジトリを提供し、そのメタデータを使用してデータのクエリと変換を行います。Lake Formation は、AWS Glue Data Catalog を使用して、データレイク、データソース、変換、およびターゲットに関するメタデータを保存します。

データソースとターゲットに関するメタデータは、データベースとテーブルの形式になっています。テーブルは、スキーマ情報、およびロケーション情報などを保存します。データベースはテーブルのコレクションです。Lake Formation は、Data Catalog 内のデータベースとテーブルへのアクセスを制御するための許可の階層を提供します。

各 AWS アカウントには、AWS リージョンごとに 1 つの Data Catalog があります。

基盤となるデータ

基礎となるデータは、データカタログのテーブルが指すソースデータまたはデータレイク内のデータを指します。

プリンシパル

プリンシパルは、AWS Identity and Access Management (IAM) のユーザーまたはロール、あるいは Active Directory のユーザーです。

データレイク管理者

データレイク管理者は、あらゆるデータカタログリソースまたはデータの場所に対するアクセス許可を任意のプリンシパル (自分自身を含む) に付与できるプリンシパルです。データレイク管理者は、Data Catalog の最初のユーザーとして指定します。このユーザーは、リソースのより詳細な許可を他のプリンシパルに付与できるようになります。

> **Note**
>
> IAM 管理ユーザー (`AdministratorAccess` AWS マネージドポリシーを持つユーザー) が、自動的にデータレイク管理者になるわけではありません。例えば、IAM 管理ユーザーがカタログオブジェクトに対する Lake Formation 許可を付与できるのは、これを実行する許可が IAM 管理ユーザー付与されている場合のみになります。ただし、IAM 管理ユーザーは、Lake Formation コンソールまたは API を使用して、自分自身をデータレイク管理者として指定できます。

データレイク管理者の能力については、「黙示的な Lake Formation 許可」を参照してください。ユーザーのデータレイク管理者としての指定については、「データレイク管理者を作成する」を参照してください。

AWS Lake Formation との サービス統合

Lake Formation を使用して、Amazon S3 に保存されているデータに対するデータベース、テーブル、および列レベルのアクセス許可を管理できます。データを Lake Formation に登録したら、Amazon Athena AWS Glue、Amazon Redshift Spectrum、Amazon EMR などの AWS 分析サービスを使用してデータをクエリできます。以下の AWS サービスは、Lake Formation のアクセス許可と統合 AWS Lake Formation され、そのアクセス許可を尊重します。

AWS サービス	統合の詳細
AWS Glue	参照トピック: AWS Lake Formation での の使用 AWS Glue AWS Glue と Lake Formation は同じ Data Catalog を共有しています。AWS Glue ユーザーがコンソール操作 (テーブルのリストの表示など) およびすべての API 操作のためにアクセスできるのは、ユーザーが Lake Formation 許可を持つデータベースとテーブルのみです。

AWS サービス	統合の詳細
Amazon Athena	参照トピック: Amazon Athena AWS Lake Formation での の使用 Lake Formation を使用して、Amazon S3 内のデータへの読み取りアクセス権を許可または拒否できます。 Amazon Athena ユーザーがクエリエディタで AWS Glue カタログを選択する場合、ユーザーは Lake Formation 許可を持つデータベース、テーブル、列のみをクエリできます。マニフェストを使用したクエリはサポートされていません。 現在、Lake Formation は、オープンテーブルフォーマットのテーブルに対する `VACUUM`、`MERGE`、`UPDATE`、`OPTIMIZE` などの書き込み操作の権限管理をサポートしていません。 Lake Formation は、 AWS Identity and Access Management (IAM) を介して Athena で認証するプリンシパルに加えて、JDBC または ODBC ドライバーを介して接続し、SAML を介して認証する Athena ユーザーをサポートしています。サポートされている SAML プロバイダーには、Okta および Microsoft Active Directory フェデレーションサービス (AD FS) などがあります。
Amazon Redshift Spectrum	参照トピック: Amazon Redshift Spectrum AWS Lake Formation での の使用 Amazon Redshift ユーザーが のデータベースに外部スキーマを作成すると AWS Glue Data Catalog、Lake Formation アクセス許可を持つスキーマ内のテーブルと列のみをクエリできます。
Amazon Quick Enterprise Edition	参照: Quick AWS Lake Formation での の使用 Amazon Quick Enterprise Edition ユーザーが Amazon S3 ロケーションのデータセットにクエリを実行する場合、ユーザーはデータに対する Lake Formation アクセス`SELECT`許可を持っている必要があります。

AWS サービス	統合の詳細
Amazon EMR	参照: Amazon EMR AWS Lake Formation での の使用 ランタイムロールを使用して Amazon EMR クラスターを作成するときに、Lake Formation のアクセス許可を統合できます。 ランタイムロールは、Amazon EMR ジョブまたはクエリに関連付ける IAM ロールであり、Amazon EMR はこのロールを使用して AWS リソースにアクセスします。

Lake Formation は AWS Key Management Service (AWS KMS) とも連動し、Amazon Simple Storage Service (Amazon S3) ロケーションにあるデータの暗号化と復号化を行うために、これらの統合サービスをより簡単にセットアップできるようにします。

追加の Lake Formation リソース

AWS Lake Formation の詳細については、以下のリソースを使用して、このガイドで紹介した概念についてさらに理解を深めることをお勧めします。

トピック

- ブログ
- テックトークとウェビナー
- 最新のアーキテクチャ
- データメッシュリソース
- ベストプラクティスガイド

ブログ

- AWS Lake Formation 2022 年を振り返って
- 耐障害性の高いマルチリージョンの最新データアーキテクチャ
- LF タグを使用して IAM プリンシパルに指示するクロスアカウント共有
- Lake Formation 許可インベントリダッシュボード
- イベント駆動型データメッシュ

テックトークとウェビナー

- re:Invent 2020 – データレイク: AWS Lake Formation と簡単に構築、保護、共有
- re:Invent 2022 – Amazon S3 でのデータレイクの構築と運用
- AWS Summit SF 2022 – 最新のデータアーキテクチャの理解と実現
- AWS Summit ATL 2022 – AWS Lake Formation、Amazon Redshift、AWS Glue を使用した最新のデータレイク
- AWS Summit ANZ 2022 – データレイク、レイクハウス、データメッシュ: 何を、なぜ、どのように?
- AWS オンラインテックトーク – データレイクにおける権限とガバナンスの簡素化

最新のアーキテクチャ

- 最新のアーキテクチャパターン

データメッシュリソース

- AWS Lake Formation タグベースのアクセス制御を使用して、最新のデータアーキテクチャとデータメッシュパターンを大規模に構築
- JPMorgan Chase がエンタープライズデータプラットフォームを強化するために大きな価値をもたらすデータメッシュアーキテクチャを構築した方法
- AWS でデータメッシュを構築

ベストプラクティスガイド

- AWS Lake Formation ベストプラクティスガイド

Lake Formation の使用の開始

以下のセクションから開始することが推奨されます。

- AWS Lake Formation: 仕組み – 重要な用語と、様々なコンポーネントが相互作用する方法を学びます。

- Lake Formation の使用の開始 – 前提条件に関する情報を入手して、重要なセットアップタスクを完了します。
- AWS Lake Formation チュートリアル – ステップバイステップのチュートリアルに従って、Lake Formation の使用方法を学びます。
- AWS Lake Formation のセキュリティ – Lake Formation でのデータへのアクセスをセキュア化する方法を理解します。

Lake Formation の使用の開始

にサインアップしていない場合、 AWS または開始するためのサポートが必要な場合は、必ず次のタスクを完了してください。

トピック

- 初期設定 AWS タスクを完了する
- セットアップ AWS Lake Formation
- AWS Lake Formation モデルへのAWS Glueデータアクセス許可のアップグレード
- AWS Lake Formation およびインターフェイス VPC エンドポイント (AWS PrivateLink)

初期設定 AWS タスクを完了する

AWS Lake Formation を使用するには、最初に以下のタスクを完了する必要があります。

トピック

- にサインアップする AWS アカウント

にサインアップする AWS アカウント

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

セットアップ AWS Lake Formation

以下のセクションでは、Lake Formation を初めて設定する場合について説明します。Lake Formation の使用を開始するにあたり、すべての設定事項が必要になるわけではありません。手順を使用して Lake Formation アクセス許可モデルを設定し、Amazon Simple Storage Service (Amazon S3) の既存の AWS Glue Data Catalog オブジェクトとデータの場所を管理できます。

1. データレイク管理者を作成する
2. デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する
3. the section called “データレイク用の Amazon S3 ロケーションを設定する”
4. the section called “Lake Formation ユーザーにアクセス許可を割り当てる”

5. the section called “IAM アイデンティティセンターの統合”
6. the section called “(オプション) 外部データフィルタリング設定”
7. the section called “(オプション) Data Catalog 暗号化キーへのアクセス権を付与する”
8. (オプション) ワークフロー用の IAM ロールを作成する

このセクションでは、Lake Formation リソースをセットアップする 2 つの異なる方法を示します。

- AWS CloudFormation テンプレートの使用
- Lake Formation コンソールの使用

AWS コンソールを使用して Lake Formation を設定するには、「」を参照してくださいデータレイク管理者を作成する。

CloudFormation テンプレートを使用して Lake Formation リソースをセットアップする

> **Note**
>
> CloudFormation スタックは、ステップ 2 と 5 を除き、上記のステップ 1 ~ 6 を実行します。Lake Formation コンソールから、手動で「デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する」および「the section called “IAM アイデンティティセンターの統合”」を実行してください。

1. 米国東部 (バージニア北部) リージョンの IAM 管理者として https://console.aws.amazon.com/cloudformation で AWS CloudFormation コンソールにサインインします。
2. [スタックの起動] を選択します。
3. [Create Stack] (スタックの作成) 画面で、[Next] (次へ) を選択します。
4. [Stack name] (スタック名) を入力します。
5. [DatalakeAdminName] と [DatalakeAdminPassword] に、データレイク管理者ユーザーとして自分のユーザーネームとパスワードを入力します。
6. [DatalakeUser1Name] と [DatalakeUser1Password] に、データレイクアナリストユーザーとして自分のユーザーネームとパスワードを入力します。
7. [DataLakeBucketName] に、作成する新しいバケットの名前を入力します。

8. [次へ] を選択します。
9. 次のページで、[I acknowledge that CloudFormation might create IAM resources with custom names] を選択し、[次へ] を選択します。
10. 最終ページの詳細を確認し、IAM リソースを作成する AWS CloudFormation 可能性があることを確認します。
11. [作成] を選択します。

    スタックの作成には、最大 2 分かかる場合があります。

リソースをクリーンアップする

CloudFormation スタックリソースをクリーンアップする場合:

1. スタックが作成し、データレイクの口ケーションとして登録した Amazon S3 バケットの登録を解除します。
2. CloudFormation スタックを削除します。これにより、スタックによって作成されたすべてのリソースが削除されます。

データレイク管理者を作成する

データレイク管理者は、最初はデータロケーションと Data Catalog リソースに対する Lake Formation アクセス許可を任意のプリンシパル AWS Identity and Access Management (自己を含む) に付与できる唯一の (IAM) ユーザーまたはロールです。データレイク管理者の能力に関する詳細については、「黙示的な Lake Formation 許可」を参照してください。Lake Formation はデフォルトで、最大 30 人のデータレイク管理者の作成を許可します。

データレイク管理者は、Lake Formation コンソール、または Lake Formation API の `PutDataLakeSettings` 操作を使用して作成できます。

データレイク管理者の作成には、以下の許可が必要です。`Administrator` ユーザーは、これらの許可を黙示的に持っています。

- `lakeformation:PutDataLakeSettings`
- `lakeformation:GetDataLakeSettings`

`AWSLakeFormationDataAdmin` ポリシーをユーザーに付与する場合、そのユーザーは追加の Lake Formation 管理者ユーザーを作成できなくなります。

データレイク管理者を作成する (コンソール)

1. データレイク管理者になるユーザーがまだ存在しない場合は、IAM コンソールを使用してそのユーザーを作成します。存在する場合は、データレイク管理者になる既存のユーザーを選択します。

   > **Note**
   >
   > データレイク管理者として IAM 管理ユーザー (`AdministratorAccess` AWS 管理ポリシーを持つユーザー) を選択しないことをお勧めします。

   次の AWS 管理ポリシーをユーザーにアタッチします。

ポリシー	必須/オプション	メモ
`AWSLakeFormationDataAdmin`	必須	基本的なデータレイク管理者許可。この AWS 管理ポリシーには、ユーザーが新しいデータレイク管理者を作成する`PutDataLakeSetting` ことを制限する Lake Formation API オペレーションの明示的な拒否が含まれています。
`AWSGlueConsoleFullAccess` , `CloudWatchLogsReadOnlyAccess`	オプションです。	データレイク管理者が Lake Formation ブループリントから作成されたワークフローをトラブルシューティングを行う場合は、これらのポリシーをアタッチします。これらのポリシーは、データレイク管理者が AWS Glue コンソールと Amazon CloudWatch Logs コンソールでトラブルシューティング情報を表示できるようにします。ワークフローについては、「the section called “ワークフローを使用したデータのインポート”」を参照してください。

ポリシー	必須/オプション	メモ
AWSLakeFormationCrossAccountManager	オプション	このポリシーをアタッチして、データレイク管理者が Data Catalog リソースに対するクロスアカウント許可の付与と取り消しを実行できるようにします。詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。
AmazonAthenaFullAccess	オプションです。	データレイク管理者がクエリを実行する場合は、このポリシーをアタッチします Amazon Athena。

2. 以下のインラインポリシーをアタッチします。これは、Lake Formation サービスリンクロールを作成する許可をデータレイク管理者に付与します。ポリシーに推奨される名前は LakeFormationSLR です。

   このサービスリンクロールは、データレイク管理者がより簡単に Amazon S3 ロケーションを Lake Formation に登録できるようにします。Lake Formation サービスリンクロールの詳細については、「the section called “サービスにリンクされたロールの使用”」を参照してください。

   > **Important**
   >
   > 次のすべてのポリシーで、*<account-id>* を有効な AWS アカウント番号に置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "lakeformation.amazonaws.com"
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::<account-id>:role/aws-service-role/
lakeformation.amazonaws.com/AWSServiceRoleForLakeFormationDataAccess"
        }
    ]
}
```

3. (オプション) 以下の PassRole インラインポリシーをユーザーにアタッチします。このポリシーは、データレイク管理者がワークフローを作成して実行できるようにします。iam:PassRole は、ワークフローが LakeFormationWorkflowRole ロールを引き受けてクローラとジョブを作成し、作成されたクローラとジョブにロールをアタッチすることを可能にします。ポリシーに推奨される名前は UserPassRole です。

> **Important**
>
> *<account-id>* を有効な AWS アカウント番号に置き換えます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PassRolePermissions",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/LakeFormationWorkflowRole"
            ]
        }
    ]
```

```
}
```

4. (オプション) アカウントがクロスアカウント Lake Formation 許可を付与または受ける場合は、この追加のインラインポリシーをアタッチします。このポリシーにより、データレイク管理者は AWS Resource Access Manager (AWS RAM) リソース共有の招待を表示して受け入れることができます。また、 AWS Organizations 管理アカウントのデータレイク管理者の場合、ポリシーには組織へのクロスアカウント許可を有効にするアクセス許可が含まれます。詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

   ポリシーに推奨される名前は `RAMAccess` です。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ram:AcceptResourceShareInvitation",
                "ram:RejectResourceShareInvitation",
                "ec2:DescribeAvailabilityZones",
                "ram:EnableSharingWithAwsOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

5. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、`AdministratorAccess` AWS 管理ポリシーを使用してユーザーとしてサインインします。
6. [Welcome to Lake Formation] (Lake Formation へようこそ) ウィンドウが表示されたら、ステップ 1 で作成または選択した IAM ユーザーを選択し、[Get started] (開始する) を選択します。
7. [Welcome to Lake Formation] (Lake Formation へようこそ) ウィンドウが表示されない場合は、以下の手順を実行して Lake Formation 管理者を設定します。

   a. ナビゲーションペインで、[管理] の [管理ロールとタスク] を選択します。コンソールページの [データレイク管理者] セクションで、[追加] を選択します。

   b. [管理者を追加] ダイアログボックスで、[アクセスタイプ] の [データレイク管理者] を選択します。

c. [IAM ユーザーおよびロール] として、ステップ 1 で作成または選択した IAM ユーザーを選択し、[保存] を選択します。

デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する

Lake Formation は、既存の AWS Glue Data Catalog 動作との互換性のために有効になっている「IAM アクセスコントロールのみを使用する」設定から始まります。この設定により、IAM ポリシーと Amazon S3 バケットポリシーを通じて、データレイク内のデータとそのメタデータへのアクセスを管理できます。

データレイクのアクセス許可を IAM および Amazon S3 モデルから Lake Formation のアクセス許可に簡単に移行できるように、Data Catalog ではハイブリッドアクセスモードを使用することをお勧めします。ハイブリッドアクセスモードを使用すると、増分パスにより、他の既存のユーザーやワークロードを中断することなく、特定のユーザーのセットに対して Lake Formation アクセス許可を有効にすることができます。

詳細については、「ハイブリッドアクセスモード」を参照してください。

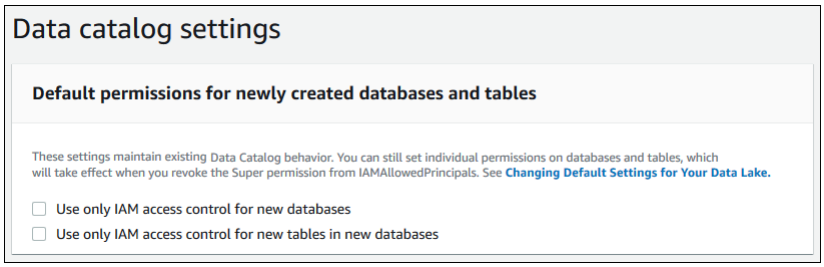
デフォルト設定を無効にすると、テーブルの既存のユーザー全員が 1 ステップで Lake Formation に移動されます。

> **Important**
>
> 既存の AWS Glue Data Catalog データベースとテーブルがある場合は、このセクションの手順を実行しないでください。その代わりに、「the section called “Lake Formation モデルに対する AWS Glue データの許可のアップグレード”」の手順を実行してください。

> **Warning**
>
> Data Catalog にデータベースとテーブルを作成するオートメーションを設定している場合、以下の手順は、オートメーションとダウンストリームの抽出、変換、ロード (ETL) ジョブが失敗する原因になる可能性があります。この手順は、既存のプロセスを変更するか、必要なプリンシパルに明示的な Lake Formation 許可を付与した後でのみ、続行するようにしてください。Lake Formation 許可については、「the section called “Lake Formation 許可のリファレンス”」を参照してください。

デフォルトの Data Catalog 設定を変更する

1. 引き続き Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を使用します。AdministratorAccess AWS 管理ポリシーでユーザーとしてサインインしていることを確認します。
2. Data Catalog 設定を変更します。
   a. ナビゲーションペインの [管理] で、[データカタログの設定] を選択します。
   b. 両方のチェックボックスをオフにして、[Save] (保存) を選択します。



3. データベース作成者の IAMAllowedPrincipals 許可を取り消します。
   a. ナビゲーションペインで、[管理] の [管理ロールとタスク] を選択します。
   b. [Administrative roles and tasks] (管理ロールとタスク) コンソールページの [Database creators] (データベース作成者) セクションで IAMAllowedPrincipals グループを選択し、[Revoke] (取り消す) を選択します。

      IAMAllowedPrincipals に [Create database] (データベースの作成) 許可があることを示す、許可の [Revoke] (取り消す) ダイアログボックスが表示されます。
   c. [Revoke] (取り消す) を選択します。

Lake Formation ユーザーにアクセス許可を割り当てる

データレイクにアクセスできるユーザーを作成します AWS Lake Formation。このユーザーは、データレイクをクエリするための最小特権アクセス許可を持っています。

ユーザーやグループの作成の詳細については、「IAM ユーザーガイド」の「IAM アイデンティティ」を参照してください。

Lake Formation データにアクセスするためのアクセス許可を管理者以外のユーザーにアタッチするには

1. で IAM コンソールを開きhttps://console.aws.amazon.com/iam、 `AdministratorAccess` AWS 管理ポリシーを使用してユーザーとしてサインインします。
2. [ユーザー] または [ユーザーグループ] を選択します。
3. 一覧から、ポリシーを埋め込むユーザーまたはグループの名前を選択します。

   [アクセス許可] を選択します。
4. [アクセス許可の追加]、[ポリシーを直接アタッチする] の順に選択します。[Filter policies] (フィルターポリシー) テキストフィールドに「`Athena`」と入力します。結果のリストで、`AmazonAthenaFullAccess` のボックスをオンにします。
5. [Create policy] (ポリシーの作成) ボタンを選択します。[ポリシーの作成] ページで、[JSON] タブを選択します。以下のコードをコピーして、ポリシーエディタに貼り付けます。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess",
                "glue:GetTable",
                "glue:GetTables",
                "glue:SearchTables",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetPartitions",
                "lakeformation:GetResourceLFTags",
                "lakeformation:ListLFTags",
                "lakeformation:GetLFTag",
                "lakeformation:SearchTablesByLFTags",
                "lakeformation:SearchDatabasesByLFTags"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

6. ポリシーの確認ページが表示されるまで次へを選択します。ポリシーの名前を入力します (`DatalakeUserBasic` など)。[ポリシーを作成] を選択し、[ポリシー] タブまたはブラウザウィンドウを閉じます。

データレイク用の Amazon S3 ロケーションを設定する

データレイク内のデータの管理とセキュア化に Lake Formation を使用するには、まず Amazon S3 ロケーションを登録する必要があります。ロケーションを登録すると、その Amazon S3 パスと、そのパスにあるすべてのフォルダが登録され、Lake Formation によるストレージレベルの許可の適用が可能になります。ユーザーが Amazon Athena などの統合エンジンからのデータをリクエストすると、Lake Formation はユーザーの許可を使用するのではなく、データアクセスを提供します。

ロケーションを登録するときは、そのロケーションに対する読み取り/書き込み許可を付与する IAM ロールを指定します。Lake Formation は、登録された Amazon S3 ロケーションのデータへのアクセスをリクエストする統合 AWS サービスに一時的な認証情報を提供するときに、そのロールを引き受けます。ユーザーは、Lake Formation サービスリンクロール (SLR) を指定するか、独自のロールを作成することができます。

カスタムロールは、以下の状況で使用します。

- Amazon CloudWatch Logs へのメトリクスの発行を計画している。ユーザー定義ロールには、SLR のアクセス許可に加えて、CloudWatch Logs でのログの追加と、メトリクスの発行のためのポリシーが含まれている必要があります。必要な CloudWatch 許可を付与するインラインポリシーの例については、「ロケーションの登録に使用されるロールの要件」を参照してください。
- Amazon S3 ロケーションが別のアカウント内に存在します。詳細については、「the section called “別の AWS アカウントでの Amazon S3 ロケーションの登録”」を参照してください。
- Amazon S3 ロケーションに AWS マネージドキーで暗号化されたデータが含まれている。詳細については、「暗号化された Amazon S3 ロケーションの登録」および「AWS アカウント間で暗号化された Amazon S3 の場所を登録する」を参照してください。
- Amazon EMR を使用して Amazon S3 ロケーションにアクセスすることを予定している。ロール要件の詳細については、「Amazon EMR 管理ガイド」の「Lake Formation 向けの IAM ロール」を参照してください。

「ロケーションの登録に使用されるロールの要件」で説明したように、選択するロールには必要な許可がある必要があります。Amazon S3 ロケーションを登録する方法の手順については、「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

(オプション) 外部データフィルタリング設定

サードパーティーのクエリエンジンを使用してデータレイク内のデータを分析および処理する予定の場合は、Lake Formation によって管理されるデータに外部エンジンがアクセスできるようにオプトインする必要があります。オプトインしない場合、外部エンジンは、Lake Formation に登録されている Amazon S3 ロケーションにあるデータにアクセスできません。

Lake Formation は、テーブル内の特定の列へのアクセスを制限するために、列レベルの許可をサポートしています。Amazon Redshift Spectrum Amazon Athena、Amazon EMR などの統合分析サービスは、 からフィルタリングされていないテーブルメタデータを取得します AWS Glue Data Catalog。クエリ応答内にある列の実際のフィルタリングは、統合サービスが担当します。データへの不正アクセスを回避するための許可の適切な処理は、サードパーティー管理者の責任になります。

サードパーティーエンジンによるデータへのアクセスとフィルタリングを許可するようにオプトインするには (コンソール)

1. 引き続き Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を使用します。Lake Formation の `PutDataLakeSettings` API 操作に対する IAM 許可を持つプリンシパルとしてサインインしていることを確認します。この許可は、「にサインアップする AWS アカウント」で作成した IAM 管理者ユーザーが持っています。
2. ナビゲーションペインの [管理] で、[アプリケーションの統合設定] を選択します。
3. [アプリケーションの統合設定] ページで、次の操作を行います。

    a. [Allow external engines to filter data in Amazon S3 locations registered with Lake Formation] (外部エンジンが、Lake Formation に登録された Amazon S3 ロケーション内のデータをフィルタリングすることを許可する) チェックボックスをオンにします。

    b. サードパーティーエンジン用に定義された [Session tag values] (セッションタグ値) を入力します。

    c. [AWS アカウント ID] に、Lake Formation に登録されているロケーションにサードパーティーのエンジンがアクセスできるアカウント ID を入力します。各アカウント ID の後で Enter キーを押します。

    d. [保存] を選択します。

セッションタグを検証せずに外部エンジンがデータにアクセスできるように方法については、「フルテーブルアクセスのためのアプリケーション統合」を参照してください。

(オプション) Data Catalog 暗号化キーへのアクセス権を付与する

AWS Glue Data Catalog が暗号化されている場合は、Data Catalog データベースとテーブルに対する Lake Formation アクセス許可を付与する必要があるすべてのプリンシパルに AWS KMS 、キーに対する AWS Identity and Access Management (IAM) アクセス許可を付与します。

詳細については、AWS Key Management Service デベロッパーガイドを参照してください。

(オプション) ワークフロー用の IAM ロールを作成する

を使用すると AWS Lake Formation、 AWS Glue クローラによって実行されるワークフローを使用してデータをインポートできます。ワークフローは、データレイクにデータをインポートするためのデータソースとスケジュールを定義します。ワークフローは、Lake Formation が提供するブループリント (テンプレート) を使用して簡単に定義できます。

ワークフローを作成するときは、Lake Formation にデータを取り込むために必要なアクセス許可を付与する AWS Identity and Access Management (IAM) ロールを割り当てる必要があります。

以下の手順では、IAM に精通していることが前提となっています。

ワークフロー用の IAM ロールを作成する

1. で IAM コンソールを開きhttps://console.aws.amazon.com/iam、 `AdministratorAccess` AWS 管理ポリシーを使用してユーザーとしてサインインします。
2. ナビゲーションペインで [Roles] (ロール)、[Create role] (ロールを作成) の順に選択します。
3. [Create role] (ロールを作成) ページで、[AWS service] (サービス) を選択して、[Glue] を選択します。[次へ] を選択します。
4. [許可の追加] ページで、AWSGlueServiceRole マネージドポリシーを検索し、リスト内のポリシー名の横にあるチェックボックスにチェックを入れます。次に、ロールに `LFWorkflowRole` という名前を付けて、[Create role] (ロールを作成) ウィザードを完了します。最後に、[Create role] (ロールを作成) を選択します。
5. [ロール] ページに戻り、`LFWorkflowRole` を検索してロール名を選択します。
6. ロールの [概要] ページにある [アクセス許可] タブで、[インラインポリシーの作成] を選択します。[ポリシーの作成] 画面で、[JSON] タブに移動し、次のインラインポリシーを追加します。ポリシーに推奨される名前は `LakeFormationWorkflow` です。

> **Important**
>
> 次のポリシーで、*<account-id>* を有効な AWS アカウント 番号に置き換えます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                 "lakeformation:GetDataAccess",
                 "lakeformation:GrantPermissions"
             ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["iam:PassRole"],
            "Resource": [
                "arn:aws:iam::111122223333:role/LakeFormationWorkflowRole"
            ]
        }
    ]
}
```

以下は、このポリシー内にある許可の簡単な説明です。

- `lakeformation:GetDataAccess` は、ワークフローによって作成されたジョブによるターゲットロケーションへの書き込みを可能にします。
- `lakeformation:GrantPermissions` は、ワークフローがターゲットテーブルに対する SELECT 許可を付与することを可能にします。
- `iam:PassRole` は、サービスが LakeFormationWorkflowRole ロールを引き受けてクローラーとジョブ (ワークフローのインスタンス) を作成し、作成されたクローラーとジョブにロールをアタッチすることを可能にします。

7. `LakeFormationWorkflowRole` ロールに 2 つのポリシーがアタッチされていることを確認します。
8. データレイクロケーションの外にあるデータを取り込んでいる場合は、そのソースデータを読み取るための許可を付与するインラインポリシーを追加します。

AWS Lake Formation モデルへのAWS Glueデータアクセス許可のアップグレード

AWS Lake Formation アクセス許可により、データレイク内のデータのきめ細かなアクセスコントロールが可能になります。Lake Formation アクセス許可モデルを使用して、Amazon Simple Storage Service (Amazon S3) の既存の AWS Glue Data Catalog オブジェクトとデータロケーションを管理できます。

Lake Formation アクセス許可モデルは、API サービスアクセスに粗粒度 AWS Identity and Access Management (IAM) アクセス許可を使用します。Lake Formation は、Lake Formation でのデータフィルタリングとセルレベルのセキュリティ 機能を使用して、ユーザーとそのアプリケーションのテーブルアクセスを列、行、セルレベルで制限します。これに対して、AWS Glue モデルはアイデンティティベースおよびリソースベースの IAM ポリシーを介してデータアクセスを許可します。

これらを切り替えるには、本ガイドの手順を実行してください。

詳細については、「Lake Formation 許可の概要」を参照してください。

デフォルトのアクセス許可について

との下位互換性を維持するためにAWS Glue、デフォルトでは、 は既存のすべての AWS Glue Data Catalog リソースに対する アクセスSuper許可を `IAMAllowedPrincipals`グループに AWS Lake Formation 付与し、IAM アクセスコントロール設定のみの使用が有効になっている場合は、新しい Data Catalog リソースに対する アクセスSuper許可を付与します。これにより、Data Catalog リソースと Amazon S3 ロケーションへのアクセスは、実質的に AWS Identity and Access Management (IAM) ポリシーのみで制御されることになります。`IAMAllowedPrincipals` グループには、IAM ポリシーによって Data Catalog オブジェクトへのアクセスを許可される IAM ユーザーとロールが含まれます。この `Super` 許可は、プリンシパルが、許可の対象であるデータベースまたはテーブルで、サポートされているすべての Lake Formation 操作を実行できるようにします。

Lake Formation を使用してデータへのアクセスの管理を開始するには、Lake Formation で既存の Data Catalog リソースのロケーションを登録するか、ハイブリッドアクセスモードを使用すること

ができます。Amazon S3 ロケーションをハイブリッドアクセスモードで登録すると、そのロケーションにあるデータベースとテーブルのプリンシパルをオプトインすることで、Lake Formation 許可を有効にできます。

データレイクのアクセス許可を IAM および Amazon S3 モデルから Lake Formation のアクセス許可に簡単に移行できるように、Data Catalog ではハイブリッドアクセスモードを使用することをお勧めします。ハイブリッドアクセスモードを使用すると、増分パスにより、他の既存のユーザーやワークロードを中断することなく、特定のユーザーのセットに対して Lake Formation アクセス許可を有効にすることができます。

詳細については、「ハイブリッドアクセスモード」を参照してください。

デフォルトの Data Catalog 設定を無効にすると、テーブルの既存のユーザー全員をワンステップで Lake Formation に移動できます。

既存の AWS Glue Data Catalog データベースとテーブルでの Lake Formation 許可の使用を開始するには、以下を実行する必要があります。

1. 各データベースとテーブルに対するユーザーの既存の IAM 許可を特定します。
2. Lake Formation でこれらの許可を複製します。
3. データが含まれる各 Amazon S3 ロケーションについて、以下を実行します。
   a. その口ケーションを参照する各 Data Catalog リソースに対する `Super` 許可を `IAMAllowedPrincipals` グループから取り消します。
   b. ロケーションを Lake Formation に登録します。
4. 既存の細粒度のアクセスコントロール IAM ポリシーをクリーンアップします。

> **Important**
>
> Data Catalog の移行プロセス中に新しいユーザーを追加するには、以前と同じように IAM で詳細な AWS Glue 許可をセットアップする必要があります。また、このセクションの説明どおりに Lake Formation でこれらの許可を複製する必要もあります。新規ユーザーが、本ガイドで説明されている粗粒度の IAM ポリシーを持っている場合は、`IAMAllowedPrincipals` に付与された `Super` 許可を持つデータベースまたはテーブルならば、どれでもリストすることができます。これらのリソースのメタデータを表示することも可能です。

このセクションの手順を実行して、Lake Formation 許可モデルにアップグレードします。

トピック

- ステップ 1: ユーザーとロールの既存の許可をリストする
- ステップ 2: 同等の Lake Formation 許可をセットアップする
- ステップ 3: Lake Formation を使用するための IAM 許可をユーザーに付与する
- ステップ 4: データストアを Lake Formation 許可モデルに切り替える
- ステップ 5: 新しい Data Catalog リソースをセキュア化する
- ステップ 6: 将来のデータレイクアクセスのための新しい IAM ポリシーをユーザーに付与する
- ステップ 7: 既存の IAM ポリシーをクリーンアップする

ステップ 1: ユーザーとロールの既存の許可をリストする

既存のAWS Glueデータベースとテーブルで AWS Lake Formation アクセス許可の使用を開始するには、まずユーザーの既存のアクセス許可を決定する必要があります。

> **Important**
>
> 開始する前に、「開始方法」のタスクを確実に完了してください。

トピック

- API 操作の使用
- の使用 AWS マネジメントコンソール
- の使用 AWS CloudTrail

API 操作の使用

AWS Identity and Access Management (IAM) ListPoliciesGrantingServiceAccess API オペレーションを使用して、各プリンシパル (ユーザーまたはロール) にアタッチされた IAM ポリシーを決定します。結果で返されたポリシーから、プリンシパルに付与されている IAM 許可を確認できます。API は、プリンシパルごとに個別に呼び出す必要があります。

Example

次の の AWS CLI 例では、ユーザー にアタッチされたポリシーを返しますglue_user1。

```
aws iam list-policies-granting-service-access --arn arn:aws:iam::111122223333:user/
glue_user1 --service-namespaces glue
```

このコマンドは、以下のような結果を返します。

```
{
    "PoliciesGrantingServiceAccess": [
        {
            "ServiceNamespace": "glue",
            "Policies": [
                {
                    "PolicyType": "INLINE",
                    "PolicyName": "GlueUserBasic",
                    "EntityName": "glue_user1",
                    "EntityType": "USER"
                },
                {
                    "PolicyType": "MANAGED",
                    "PolicyArn": "arn:aws:iam::aws:policy/AmazonAthenaFullAccess",
                    "PolicyName": "AmazonAthenaFullAccess"
                }
            ]
        }
    ],
    "IsTruncated": false
}
```

の使用 AWS マネジメントコンソール

この情報は、ユーザーまたはロールの概要ページの Access Advisor タブにある AWS Identity and Access Management (IAM) コンソールでも確認できます。

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. IAM ナビゲーションペインで、[Users] (ユーザー) または [Roles] (ロール) を選択します。
3. リスト内の名前を選択すると、その [Summary] (概要) ページが表示されるので、[Access Advisor] (アクセスアドバイザー) タブを選択します。
4. 各ポリシーを調べて、各ユーザーが許可を持っているデータベース、テーブル、およびアクションの組み合わせを特定します。

データ処理ジョブがデータにアクセスするためのロールを引き受けている可能性があるため、このプロセスでは、ユーザーに加えてロールも調べるようにしてください。

の使用 AWS CloudTrail

既存のアクセス許可を判断するもう 1 つの方法は、ログの `additionaleventdata`フィールドに `insufficientLakeFormationPermissions`エントリが含まれている AWS CloudTrail AWS Glue API コールを探すことです。このエントリは、ユーザーが同じアクションを実行するために Lake Formation 許可を必要とするデータベースとテーブルをリストします。

これらはデータアクセスログであるため、ユーザーとその許可の包括的なリストを生成することは限りません。ユーザーのデータアクセスパターンを取得するには、幅広い時間範囲 (数週間または数か月など) を選択することをお勧めします。

詳細については、「AWS CloudTrail ユーザーガイド」の「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。

次は、AWS Glue 許可に相当する Lake Formation 許可をセットアップできます。「ステップ 2: 同等の Lake Formation 許可をセットアップする」を参照してください。

ステップ 2: 同等の Lake Formation 許可をセットアップする

で収集された情報を使用してステップ 1: ユーザーとロールの既存の許可をリストする、 AWS Lake Formation アクセス許可と一致するAWS Glueアクセス許可を付与します。以下の方式のいずれかを使用して、付与を実行します。

- Lake Formation コンソールまたは AWS CLIを使用する。

  「the section called “データアクセス許可の付与”」を参照してください。

- `GrantPermissions` または `BatchGrantPermissions` API 操作を使用する。

  「許可 API」を参照してください。

詳細については、「Lake Formation 許可の概要」を参照してください。

Lake Formation 許可を設定したら、「ステップ 3: Lake Formation を使用するための IAM 許可をユーザーに付与する」に進みます。

ステップ 3: Lake Formation を使用するための IAM 許可をユーザーに付与する

アクセス AWS Lake Formation 許可モデルを使用するには、プリンシパルに Lake Formation APIs に対する AWS Identity and Access Management (IAM) アクセス許可が必要です。

IAM で以下のポリシーを作成して、データレイクへのアクセス権を必要とするすべてのユーザーにポリシーをアタッチします。ポリシーには `LakeFormationDataAccess` という名前を付けます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LakeFormationDataAccess",
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

次に、Lake Formation 許可へのアップグレードを 1 度に 1 データロケーションずつ実行します。「ステップ 4: データストアを Lake Formation 許可モデルに切り替える」を参照してください。

ステップ 4: データストアを Lake Formation 許可モデルに切り替える

Lake Formation 許可へのアップグレードを 1 度に 1 データロケーションずつ実行します。これを行うには、Data Catalog によって参照されるすべての Amazon Simple Storage Service (Amazon S3) パスを登録するまで、このセクション全体を繰り返します。

トピック

- Lake Formation 許可を検証する
- 既存の Data Catalog リソースをセキュア化する

- Amazon S3 ロケーションの Lake Formation 許可を有効にする

Lake Formation 許可を検証する

ロケーションを登録する前に、検証ステップを実行して、正しいプリンシパルに必要な Lake Formation 許可があること、および Lake Formation 許可がそれらを持つべきではないプリンシパルに付与されていないことを確認します。Lake Formation `GetEffectivePermissionsForPath` API 操作を使用して、Amazon S3 ロケーションを参照する Data Catalog リソースと、これらのリソースに対する許可を持つプリンシパルを特定します。

次の の AWS CLI 例では、Amazon S3 バケット を参照する Data Catalog データベースとテーブルを返します`products`。

```
aws lakeformation get-effective-permissions-for-path --resource-arn
 arn:aws:s3:::products --profile datalake_admin
```

`profile` オプションに注意してください。このコマンドは、データレイク管理者として実行することをお勧めします。

以下は、返された結果の抜粋です。

```
{
        "PermissionsWithGrantOption": [
            "SELECT"
        ],
        "Resource": {
            "TableWithColumns": {
                "Name": "inventory_product",
                "ColumnWildcard": {},
                "DatabaseName": "inventory"
            }
        },
        "Permissions": [
            "SELECT"
        ],
        "Principal": {
            "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
datalake_user1",
            "DataLakePrincipalType": "IAM_USER"
        }
 },...
```

> **Important**
>
> AWS Glue Data Catalog が暗号化されている場合、`GetEffectivePermissionsForPath` は、Lake Formation の一般提供後に作成または変更されたデータベースとテーブルのみを返します。

既存の Data Catalog リソースをセキュア化する

次に、その口ケーションについて特定した各テーブルと各データベースに対する `Super` 許可を `IAMAllowedPrincipals` から取り消します。

> **Warning**
>
> Data Catalog にデータベースとテーブルを作成するオートメーションを設定している場合、以下の手順は、オートメーションとダウンストリームの抽出、変換、ロード (ETL) ジョブが失敗する原因になる可能性があります。この手順は、既存のプロセスを変更するか、必要なプリンシパルに明示的な Lake Formation 許可を付与した後でのみ、続行するようにしてください。Lake Formation 許可については、「the section called “Lake Formation 許可のリファレンス”」を参照してください。

テーブルに対する **`Super`** を **`IAMAllowedPrincipals`** から取り消す

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者としてサインインします。
2. ナビゲーションペインで [Table] (テーブル) を選択します。
3. [Tables] (テーブル) ページで、目的のテーブルの横にあるラジオボタンを選択します。
4. [Actions] (アクション) メニューで、[Revoke] (取り消す) を選択します。
5. [Revoke permissions] (許可を取り消す) ダイアログボックスの [IAM users and roles] (IAM ユーザーおよびロール) リストで、[Group]グループ見出しまでスクロールダウンして [IAMAllowedPrincipals] を選択します。
6. [Table permissions] (テーブルの許可) で [Super] (スーパー) が選択されていることを確認してから、[Revoke] (取り消す) を選択します。

データベースに対する **Super** を **IAMAllowedPrincipals** から取り消す

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者としてサインインします。
2. ナビゲーションペインで、[Databases] (データベース) を選択します。
3. [Databases] (データベース) ページで、目的のデータベースの横にあるラジオボタンを選択します。
4. [Actions] (アクション) メニューで、[Edit] (編集) を選択します。
5. [Edit database] (データベースの編集) ページで、[Use only IAM access control for new tables in this database] (このデータベースの新しいテーブルには IAM アクセスコントロールのみを使用する) をオフにしてから [Save] (保存) を選択します。
6. [Databases] (データベース) ページに戻り、データベースが選択されていることを確認してから、[Actions] (アクション) メニューで [Revoke] (取り消す) を選択します。
7. [Revoke permissions] (許可を取り消す) ダイアログボックスの [IAM users and roles] (IAM ユーザーおよびロール) リストで、[Group]グループ見出しまでスクロールダウンして [IAMAllowedPrincipals] を選択します。
8. [Database permissions] (データベースの許可) で [Super] (スーパー) が選択されていることを確認してから、[Revoke] (取り消す) を選択します。

Amazon S3 ロケーションの Lake Formation 許可を有効にする

次に、Amazon S3 ロケーションを Lake Formation に登録します。これを実行するには、「データレイクへの Amazon S3 ロケーションの追加」で説明されているプロセスを使用できます。または、「認証情報供給 API」の説明に従って `RegisterResource` API 操作を使用します。

> **Note**
>
> 親ロケーションが登録されている場合、子ロケーションを登録する必要はありません。

これらの手順を完了して、ユーザーがそのデータにアクセスできることをテストしたら、Lake Formation 許可を正常にアップグレードしたことになります。次のステップである「ステップ 5: 新しい Data Catalog リソースをセキュア化する」に進みます。

ステップ 5: 新しい Data Catalog リソースをセキュア化する

次に、デフォルト Data Catalog 設定を変更することによって、すべての新しい Data Catalog リソースをセキュア化します。新しいデータベースとテーブルに対してのみ AWS Identity and Access Management (IAM) アクセスコントロールを使用するオプションをオフにします。

> **Warning**
>
> Data Catalog にデータベースとテーブルを作成するオートメーションを設定している場合、以下の手順は、オートメーションとダウンストリームの抽出、変換、ロード (ETL) ジョブが失敗する原因になる可能性があります。この手順は、既存のプロセスを変更するか、必要なプリンシパルに明示的な Lake Formation 許可を付与した後でのみ、続行するようにしてください。Lake Formation 許可については、「the section called “Lake Formation 許可のリファレンス”」を参照してください。

デフォルトの Data Catalog 設定を変更する

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。IAM 管理ユーザー (ユーザー`Administrator`または `AdministratorAccess` AWS 管理ポリシーを持つ別のユーザー) としてサインインします。
2. ナビゲーションペインで [Settings] (設定) を選択します。
3. [Data catalog settings] (Data Catalog の設定) ページで、両方のチェックボックスをオフにしてから [Save] (保存) を選択します。

次のステップは、将来追加されるデータベースまたはテーブルに対するアクセス権のユーザーへの付与です。「ステップ 6: 将来のデータレイクアクセスのための新しい IAM ポリシーをユーザーに付与する」を参照してください。

ステップ 6: 将来のデータレイクアクセスのための新しい IAM ポリシーをユーザーに付与する

今後、追加の Data Catalog データベースまたはテーブルへのアクセス権をユーザーに付与するには、以下の粗粒度 AWS Identity and Access Management (IAM) インラインポリシーをユーザーに付与する必要があります。ポリシーには `GlueFullReadAccess` という名前を付けます。

> **Important**
>
> Data Catalog 内のすべてのデータベースとテーブルに対する Super を IAMAllowedPrincipals から取り消す前にこのポリシーをユーザーにアタッチすると、そのユーザーは、Super が IAMAllowedPrincipals に付与されている任意のリソースのすべてのメタデータを表示することができます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "GlueFullReadAccess",
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess",
                "glue:GetTable",
                "glue:GetTables",
                "glue:SearchTables",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetPartitions"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> このステップ、および前の手順で指定されているインラインポリシーには、最小限の IAM 許可が含まれています。データレイク管理者、データアナリスト、その他のペルソナに対する推奨ポリシーについては、「the section called “Lake Formation のペルソナと IAM 許可のリファレンス”」を参照してください。

次に、「ステップ 7: 既存の IAM ポリシーをクリーンアップする」に進みます。

ステップ 7: 既存の IAM ポリシーをクリーンアップする

アクセス AWS Lake Formation 許可を設定し、粗粒度のアクセスコントロール AWS Identity and Access Management (IAM) ポリシーを作成してアタッチしたら、次のステップを完了します。

- Lake Formation で複製した古い細粒度のアクセスコントロール IAM ポリシーを、ユーザー、グループ、およびロールから削除します。

そうすることによって、これらのプリンシパルが Amazon Simple Storage Service (Amazon S3) 内のデータに直接アクセスできないことを確実にします。その後、これらのプリンシパルのデータレイクアクセスを Lake Formation を通じて完全に管理することができます。

AWS Lake Formation およびインターフェイス VPC エンドポイント (AWS PrivateLink)

Amazon VPC は、定義した仮想ネットワークで AWS リソースを起動するために使用できる AWS サービスです。VPC を使用することで、IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどのネットワーク設定を制御できます。

Amazon Virtual Private Cloud (Amazon VPC) を使用して AWS リソースをホストする場合は、VPC と Lake Formation の間にプライベート接続を確立できます。この接続を使用して、Lake Formation がパブリックインターネットを経由せずに VPC 内のリソースと通信できるようにします。

VPC と の間にプライベート接続を確立するには、インターフェイス VPC エンドポイント AWS Lake Formation を作成します。インターフェイスエンドポイントは、インターネットゲートウェイ、NAT デバイス、VPN 接続、または Direct Connect 接続を必要とせずに、Lake Formation API にプライベートにアクセスできるようにするテクノロジーである AWS PrivateLink を活用しています。VPC 内のインスタンスが Lake Formation API と通信するために パブリック IP アドレスは必要ありません。VPC と Lake Formation 間のトラフィックが Amazon ネットワークを離れることはありません。

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

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

Lake Formation VPC エンドポイントに関する考慮事項

Lake Formation のインターフェイス VPC エンドポイントをセットアップする前に、「Amazon VPC ユーザーガイド」で「インターフェイスエンドポイントのプロパティと制限」を確認するようにしてください。

Lake Formation は、その API アクションのすべてに対する VPC からの呼び出しをサポートしています。Lake Formation と Amazon VPC エンドポイントの両方 AWS リージョン をサポートするすべての で、VPC エンドポイントで Lake Formation を使用できます。

Lake Formation 用のインターフェイス VPC エンドポイントの作成

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

Lake Formation 用の VPC エンドポイントは、以下のサービス名を使用して作成します。

- com.amazonaws.*region*.lakeformation

エンドポイントに対してプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 (`lakeformation.us-east-1.amazonaws.com` など) を使用して、Lake Formation への API リクエストを実行できます。

詳細については、「Amazon VPC ユーザーガイド」の「インターフェイスエンドポイント経由でのサービスへのアクセス」を参照してください。

Lake Formation 用の VPC エンドポイントポリシーの作成

Lake Formation は VPC エンドポイントポリシーをサポートします。エンドポイントポリシーは、VPC エンドポイントにアタッチして、エンドポイントを使用して AWS サービスにアクセスできる AWS プリンシパルを制御するリソースベースのポリシーです。

VPC エンドポイントに、Lake Formation へのアクセスをコントロールするエンドポイントポリシーをアタッチできます。このポリシーは、以下の情報を指定します。

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

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

例: Lake Formation アクション用の VPC エンドポイントポリシー

以下の Lake Formation 用の VPC エンドポイントポリシー例は、Lake Formation 許可を使用した認証情報供給を許可します。このポリシーを使用して、Amazon Redshift クラスターまたはプライベートサブネットにある Amazon EMR クラスターからの Lake Formation アクセス許可を使用してクエリを実行できます。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "lakeformation:GetDataAccess",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

> **Note**
>
> エンドポイント作成時にポリシーをアタッチしない場合は、サービスへのフルアクセスを許可するデフォルトのポリシーがアタッチされます。

詳細については、Amazon VPC ドキュメントのこれらのトピックを参照してください。

- Amazon VPC とは?
- インターフェイスエンドポイントの作成
- VPC エンドポイントポリシーを使用する

AWS Lake Formation チュートリアル

これらのチュートリアルは 3 つのカテゴリに分かれており、 AWS Lake Formationを使用してデータレイクの構築とデータの取り込み、データの共有、データレイクの保護を行う方法についてステップバイステップの手順を紹介します。

1. データレイクを構築してデータを取り込む: データレイクを構築し、ブループリントを使用してデータを移動、保存、分類、消去、整理する方法について学習します。また、管理対象テーブルを設定する方法についても学習します。管理対象テーブルは、新しい Amazon S3 テーブルタイプであり、アトミック性、一貫性、分離性、耐久性 (ACID: Atomic, Consistent, Isolated, and Durable) を備えたトランザクションをサポートします。

   開始する前に、必ず Lake Formation の使用の開始 のステップを完了してください。

   - AWS CloudTrail ソースからのデータレイクの作成

     独自の CloudTrail ログをデータソースとして使用し、最初のデータレイクを作成して読み込みます。

   - Lake Formation での JDBC ソースからのデータレイクの作成

     データレイクを作成するには、リレーショナルデータベースなどの JDBC アクセス可能なデータストアの 1 つをデータソースとして使用します。

2. データレイクを保護する: タグベースおよび行レベルのアクセスコントロールを使用して、データレイクへのアクセスを効果的に保護および管理する方法について学習します。

   - Lake Formation でのオープンテーブルストレージフォーマットのアクセス許可の設定

     このチュートリアルでは、Lake Formation でオープンソースのトランザクションテーブル形式 (Apache Iceberg、Apache Hudi、Linux Foundation Delta Lake テーブル) のアクセス許可を設定する方法を示します。

   - Lake Formation のタグベースのアクセスコントロールを使用したデータレイクの管理

     Lake Formation のタグベースのアクセスコントロールを使用して、データレイク内のデータへのアクセスを管理する方法について学習します。

   - 行レベルのアクセスコントロールによるデータレイクの保護

     Lake Formation で行レベルの許可を設定し、データコンプライアンスおよびガバナンスポリシーに基づいて、特定の行へのアクセスを制限する方法について学習します。

3. データを共有する: タグベースのアクセスコントロール (TBAC) を使用して、 AWS アカウント 間でデータを安全に共有する方法と、 AWS アカウント間で共有するデータセットへのきめ細かな許可を管理する方法について学習します。

   - Lake Formation のタグベースのアクセスコントロールと名前付きリソースを使用したデータレイクの共有

     このチュートリアルでは、Lake Formation を使用して AWS アカウント 間でデータを安全に共有する方法について学習します。

   - Lake Formation のきめ細かなアクセスコントロールを使用したデータレイクの共有

     このチュートリアルでは、複数の を管理するときに Lake Formation を使用してデータセットをすばやく簡単に共有する方法について説明します AWS アカウント AWS Organizations。

トピック

- AWS CloudTrail ソースからのデータレイクの作成
- Lake Formation での JDBC ソースからのデータレイクの作成
- Lake Formation でのオープンテーブルストレージフォーマットのアクセス許可の設定
- Lake Formation のタグベースのアクセスコントロールを使用したデータレイクの管理
- 行レベルのアクセスコントロールによるデータレイクの保護
- Lake Formation のタグベースのアクセスコントロールと名前付きリソースを使用したデータレイクの共有
- Lake Formation のきめ細かなアクセスコントロールを使用したデータレイクの共有

AWS CloudTrail ソースからのデータレイクの作成

このチュートリアルでは、Lake Formation コンソールで AWS CloudTrail ソースから最初のデータレイクを作成してロードするためのアクションについて説明します。

データレイクを作成するための大まかなステップ

1. Amazon Simple Storage Service (Amazon S3) パスをデータレイクとして登録します。
2. Lake Formation に、Data Catalog、およびデータレイク内の Amazon S3 ロケーションに書き込みを行うための許可を付与します。
3. Data Catalog 内のメタデータテーブルを整理するためのデータベースを作成します。

4. ブループリントを使用してワークフローを作成します。ワークフローを実行して、データソースからデータを取り込みます。
5. 他のユーザーが Data Catalog とデータレイク内のデータを管理できるようにする Lake Formation 許可を設定します。
6. Amazon S3 データレイクにインポートしたデータをクエリするように Amazon Athena をセットアップします。
7. 一部のデータストアタイプについては、Amazon S3 データレイクにインポートしたデータをクエリするように Amazon Redshift Spectrum をセットアップします。

トピック

- 対象者
- 前提条件
- ステップ 1: データアナリストユーザーの作成
- ステップ 2: ワークフローロールに AWS CloudTrail ログを読み取るアクセス許可を追加する
- ステップ 3: データレイクとしての Amazon S3 バケットを作成する
- ステップ 4: Amazon S3 パスを登録する
- ステップ 5: データのロケーションの許可を付与する
- ステップ 6: Data Catalog でデータベースを作成する
- ステップ 7: データの許可を付与する
- ステップ 8: ブループリントを使用してワークフローを作成する
- ステップ 9: ワークフローを実行する
- ステップ 10: テーブルに対する SELECT を付与する
- ステップ 11: を使用してデータレイクをクエリする Amazon Athena

対象者

次の表は、このチュートリアルでデータレイクを作成するために使用しているロールのリストです。

対象者

ロール	説明
IAM 管理者	AWS 管理ポリシーがあります: `AdministratorAccess` 。IAM ロールと Amazon S3 バケットを作成できます。
データレイク管理者	Data Catalog へのアクセス、データベースの作成、および他のユーザーへの Lake Formation 許可の付与を実行できるユーザー。IAM 許可の数は IAM 管理者よりも少ないですが、データレイクを管理するには十分な許可を持っています。
データアナリスト	データレイクに対してクエリを実行できるユーザー。クエリを実行するために十分な許可のみを持っています。
ワークフローロール	ワークフローを実行するために必要な IAM ポリシーを持つロール。詳細については、「(オプション) ワークフロー用の IAM ロールを作成する」を参照してください。

前提条件

開始する前に、以下を確認してください。

- 「セットアップ AWS Lake Formation」のタスクを完了していること。
- CloudTrail ログのロケーションを把握していること。
- Athena では、データアナリストペルソナが Athena を使用する前に、クエリ結果を保存するための Amazon S3 バケットを作成する必要があります。

AWS Identity and Access Management (IAM) に精通していることが前提です。IAM については、「IAM ユーザーガイド」を参照してください。

ステップ 1: データアナリストユーザーの作成

このユーザーは、データレイクをクエリするための最小限の許可セットを持っています。

1. IAM コンソール (https://console.aws.amazon.com/iam) を開きます。AdministratorAccess AWS 管理ポリシーを使用してユーザーとしてサインインします。
2. 以下の設定で、datalake_user という名前のユーザーを作成します。
   - AWS マネジメントコンソール アクセスを有効にします。
   - パスワードを設定して、パスワードのリセットを不要にする。
   - AmazonAthenaFullAccess AWS 管理ポリシーをアタッチします。
   - 以下のインラインポリシーをアタッチする。ポリシーには DatalakeUserBasic という名前を付けます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess",
                "glue:GetTable",
                "glue:GetTables",
                "glue:SearchTables",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetPartitions",
                "lakeformation:GetResourceLFTags",
                "lakeformation:ListLFTags",
                "lakeformation:GetLFTag",
                "lakeformation:SearchTablesByLFTags",
                "lakeformation:SearchDatabasesByLFTags"
           ],
            "Resource": "*"
        }
    ]
}
```

ステップ 2: ワークフローロールに AWS CloudTrail ログを読み取るアクセス許可を追加する

1. 以下のインラインポリシーを LakeFormationWorkflowRole ロールにアタッチします。このポリシーは、 AWS CloudTrail ログを読み取るアクセス許可を付与します。ポリシーには DatalakeGetCloudTrail という名前を付けます。

   LakeFormationWorkflowRole ロールを作成するには、「(オプション) ワークフロー用の IAM ロールを作成する」を参照してください。

   > **Important**
   >
   > *<your-s3-cloudtrail-bucket>* は、CloudTrail データの Amazon S3 ロケーションに置き換えてください。

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "s3:GetObject",
               "Resource": ["arn:aws:s3:::<your-s3-cloudtrail-bucket>/*"]
           }
       ]
   }
   ```

2. ロールに 3 つのポリシーがアタッチされていることを確認します。

ステップ 3: データレイクとしての Amazon S3 バケットを作成する

データレイクのルートロケーションになる Amazon S3 バケットを作成します。

1. https://console.aws.amazon.com/s3/ で Amazon S3 コンソールを開き、管理者権限を持つユーザーとしてサインインします。

2. [Create bucket] (バケットを作成) を選択し、ウィザードをすべて実行して *<yourName>*-datalake-cloudtrail という名前のバケットを作成します。*<yourName>* はユーザーの名前のイニシャルと苗字の組み合わせです。例: jdoe-datalake-cloudtrail。

   Amazon S3 バケットの詳しい作成手順については、「バケットの作成」を参照してください。

ステップ 4: Amazon S3 パスを登録する

Amazon S3 パスをデータレイクのルートロケーションとして登録します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。データレイク管理者としてサインインします。
2. ナビゲーションペインの [Register and ingest] (登録および取り込み) で [Data lake locations] (データレイクのロケーション) を選択します。
3. [Register location] (ロケーションを登録) を選択してから、[Browse] (参照) を選択します。
4. 前に作成した *<yourName>*-datalake-cloudtrail バケットを選択し、デフォルトの IAM ロール AWSServiceRoleForLakeFormationDataAccess を受け入れ、[Register location] (ロケーションを登録) を選択します。

   ロケーションの登録に関する詳細については、「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

ステップ 5: データのロケーションの許可を付与する

プリンシパルは、作成する Data Catalog のテーブルまたはデータベースのポイント先となるデータレイクロケーションに対するデータロケーション許可を持っている必要があります。ワークフローの IAM ロールにデータロケーション許可を付与して、ワークフローがデータ取り込み先に書き込みを実行できるようにする必要があります。

1. ナビゲーションペインの [Permissions] (許可) で [Data locations] (データのロケーション) を選択します。
2. [Grant] (付与) を選択し、[Grant permissions] (許可の付与) ダイアログボックスで、以下の選択を行います。

   a. [IAM user and roles] (IAM ユーザーおよびロール) で、LakeFormationWorkflowRole を選択します。

b. [Storage locations] (ストレージのロケーション) で、使用する *<yourName>*-datalake-cloudtrail バケットを選択します。

3. [Grant] (付与) を選択します。

データロケーション許可については、「Underlying data access control」を参照してください。

ステップ 6: Data Catalog でデータベースを作成する

Lake Formation Data Catalog のメタデータテーブルは、データベース内に保存されます。

1. ナビゲーションペインの [Data catalog] で [Databases] (データベース) を選択します。
2. [Create database] (データベースを作成) を選択し、[Database details] (データベースの詳細) で lakeformation_cloudtrail という名前を入力します。
3. 他のフィールドは空欄のままにしておき、[Create database] (データベースを作成) を選択します。

ステップ 7: データの許可を付与する

Data Catalog でメタデータテーブルを作成するための許可を付与する必要があります。ワークフローは LakeFormationWorkflowRole ロールを使用して実行されるため、これらの許可をロールに付与する必要があります。

1. Lake Formation コンソールのナビゲーションペインにある [Data catalog] で [Databases] (データベース) を選択します。
2. lakeformation_cloudtrail データベースを選択してから、[Actions] (アクション) ドロップダウンリストで、[Permissions] (許可) の見出しの下にある [Grant] (付与) を選択します。
3. [Grant data permissions] (データ許可の付与) ダイアログボックスで、以下の選択を行います。

   a. [Principals] (プリンシパル) の [IAM user and roles] (IAM ユーザーおよびロール) で LakeFormationWorkflowRole を選択します。
   b. [LF タグまたはカタログリソース] で、[名前付きのデータカタログリソース] を選択します。
   c. [Databases] (データベース) については、lakeformation_cloudtrail データベースがすでに追加されていることが確認できるはずです。
   d. [データベースのアクセス許可] で、[テーブルを作成]、[変更]、および [ドロップ] をオンにして、[スーパー] が選択されている場合はそれをオフにします。

4. [Grant] (付与) を選択します。

Lake Formation 許可の付与に関する詳細ついては、「Lake Formation 許可の管理」を参照してください。

ステップ 8: ブループリントを使用してワークフローを作成する

CloudTrail ログを読み、その構造を理解し、データカタログで適切なテーブルを作成するには、AWS Glueクローラ、ジョブ、トリガー、ワークフローで構成されるワークフローを設定する必要があります。Lake Formation のブループリントを使用すると、このプロセスが容易になります。

ワークフローは、データを検出してデータレイクに取り込むジョブ、クローラ、およびトリガーを生成します。ワークフローは、事前定義された Lake Formation ブループリントのいずれかに基づいて作成します。

1. Lake Formation コンソールのナビゲーションペインで、取り込みでブループリントを選択し、ブループリントの使用を選択します。
2. [Use a blueprint] (ブループリントの使用) ページの [Blueprint type] (ブループリントタイプ) で [AWS CloudTrail]を選択します。
3. [Import source] (インポートソース) で、CloudTrail ソースと開始日を選択します。
4. [Import target] (インポートターゲット) で、以下のパラメータを指定します。

[Target database] (ターゲットデータベース)	`lakeformation_cloudtrail`
[Target storage location] (ターゲットストレージロケーション)	`s3://<yourName> -datalake-cloudtrail`
[Data format] (データ形式)	Parquet

5. [Import Frequency] (インポート頻度) には、[Run on demand] (オンデマンドで実行) を選択します。
6. [Import target] (インポートオプション) で、以下のパラメータを指定します。

[Workflow name] (ワークフロー名)	`lakeformationcloudtrailtest`
[IAM role] (IAM ロール)	`LakeFormationWorkflowRole`

[Table prefix] (テーブルプレフィックス)	`cloudtrailtest` **Note** 小文字を使用する必要があります。

7. [Create] (作成) を選択し、ワークフローが正常に作成されたことコンソールが報告するまで待機します。

> **Tip**
> 以下のエラーメッセージが表示されましたか?
> `User: arn:aws:iam::<account-id>:user/<datalake_administrator_user> is not authorized to perform: iam:PassRole on resource:arn:aws:iam::<account-id>:role/LakeFormationWorkflowRole...`
> その場合、データレイク管理者ユーザーのインラインポリシーで *<account-id>* を有効な AWS アカウント番号に置き換えたことを確認します。

ステップ 9: ワークフローを実行する

ワークフローがオンデマンドで実行されることを指定したので、ワークフローは手動で開始する必要があります。

- [Blueprints] (ブループリント) ページでワークフロー `lakeformationcloudtrailtest` を選択し、[Actions] (アクション) メニューから [Start] (開始) を選択します。

  ワークフローの実行に伴って、その進捗状況を [Last run status] (最終実行ステータス) 列で確認できます。更新ボタンを随時選択します。

  ステータスは、[RUNNING] (実行中) から、[Discovering] (検出中)、[Importing] (インポート中)、[COMPLETED] (完了) と移行します。

  ワークフローが完了すると、以下のようになります。

  - Data Catalog に新しいメタデータテーブルがある。
  - CloudTrail ログがデータレイクに取り込まれる。

ワークフローが失敗する場合は、以下を実行します。

a. ワークフローを選択し、[Actions] (アクション) メニューで [View graph] (グラフを表示) を選択します。

   AWS Glue コンソールでワークフローが開きます。

b. そのワークフローが選択されていることを確認し、[History] (履歴) タブを選択します。

c. [History] (履歴) で、最新の実行を選択し、[View run details] (実行の詳細を表示) を選択します。

d. 動的 (ランタイム) グラフで失敗したジョブまたはクローラを選択し、エラーメッセージを確認します。障害が発生したノードは赤色または黄色のいずれかになっています。

ステップ 10: テーブルに対する SELECT を付与する

テーブルがポイントするデータをデータアナリストがクエリできるように、新しい Data Catalog テーブルに対する SELECT 許可を付与する必要があります。

> **Note**
>
> ワークフローは、ワークフローが作成するテーブルに対する SELECT 許可を、ワークフローを実行したユーザーに自動的に付与します。このワークフローはデータレイク管理者が実行したので、データアナリストに SELECT を付与する必要があります。

1. Lake Formation コンソールのナビゲーションペインにある [Data catalog] で [Databases] (データベース) を選択します。
2. `lakeformation_cloudtrail` データベースを選択してから、[Actions] (アクション) ドロップダウンリストで、[Permissions] (許可) の見出しの下にある [Grant] (付与) を選択します。
3. [Grant data permissions] (データ許可の付与) ダイアログボックスで、以下の選択を行います。

   a. [Principals] (プリンシパル) の [IAM user and roles] (IAM ユーザーおよびロール) で `datalake_user` を選択します。

   b. [LF-Tags or catalog resources] (LF タグまたはカタログリソース) で、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。

c. [Databases] (データベース) については、`lakeformation_cloudtrail` データベースがすでに選択されているはずです。

d. [Tables] (テーブル) には `cloudtrailtest-cloudtrail` を選択します。

e. [Table and column permissions] (テーブルと列の許可) で [Select] (選択) をオンにします。

4. [Grant] (付与) を選択します。

次のステップは、データアナリストとして実行します。

ステップ 11: を使用してデータレイクをクエリする Amazon Athena

Amazon Athena コンソールを使用して、データレイク内の CloudTrail データをクエリします。

1. Athena コンソール (https://console.aws.amazon.com/athena/) を開き、データアナリストのユーザー `datalake_user` としてサインインします。
2. 必要に応じて [Get Started] (開始する) を選択して、Athena クエリエディタに進みます。
3. [Data source] (データソース) で [AwsDataCatalog] を選択します。
4. [Database] (データベース) で、`lakeformation_cloudtrail` を選択します。

   [Tables] (テーブル) リストが表示されます。
5. テーブル `cloudtrailtest-cloudtrail` の横にあるオーバーフローメニュー (縦方向に並んだ 3 つの点) で、[Preview table] (表をプレビュー)、[Run] (実行) の順に選択します。

   クエリが実行され、10 行のデータが表示されます。

   これまで Athena を使用したことがないという場合は、最初に Athena コンソールでクエリ結果を保存するための Amazon S3 ロケーションを設定する必要があります。`datalake_user` は、ユーザーが選択した Amazon S3 バケットへのアクセスに必要な許可を持っている必要があります。

> **Note**
>
> チュートリアルが完了したところで、次は組織内のプリンシパルにデータ許可とデータロケーション許可を付与します。

Lake Formation での JDBC ソースからのデータレイクの作成

このチュートリアルでは、 AWS Lake Formation コンソールで Lake Formation を使用して JDBC ソースから最初のデータレイクを作成してロードする手順について説明します。

トピック

- 対象者
- JDBC チュートリアルの前提条件
- ステップ 1: データアナリストユーザーの作成
- ステップ 2: AWS Glue で接続を作成する
- ステップ 3: データレイク用の Amazon S3 バケットを作成する
- ステップ 4: Amazon S3 パスを登録する
- ステップ 5: データのロケーションに対する許可を付与する
- ステップ 6: Data Catalog でデータベースを作成する
- ステップ 7: データの許可を付与する
- ステップ 8: ブループリントを使用してワークフローを作成する
- ステップ 9: ワークフローを実行する
- ステップ 10: テーブルに対する SELECT を付与する
- ステップ 11: を使用してデータレイクをクエリする Amazon Athena
- ステップ 12: Amazon Redshift Spectrum を使用してデータレイク内のデータをクエリする
- ステップ 13: Amazon Redshift Spectrum を使用して Lake Formation 許可を付与または取り消す

対象者

次の表は、この AWS Lake Formation JDBC チュートリアルで使用するロールのリストです。

ロール	説明
IAM 管理者	AWS Identity and Access Management (IAM) ユーザーとロール、および Amazon Simple Storage Service (Amazon S3) バケットを作成できるユーザー。`AdministratorAccess` AWS 管理ポリシーがあります。

ロール	説明
データレイク管理者	Data Catalog へのアクセス、データベースの作成、および他のユーザーへの Lake Formation 許可の付与を実行できるユーザー。IAM 許可の数は IAM 管理者よりも少ないですが、データレイクを管理するには十分な許可を持っています。
データアナリスト	データレイクに対してクエリを実行できるユーザー。クエリを実行するために十分な許可のみを持っています。
ワークフローロール	ワークフローを実行するために必要な IAM ポリシーを持つロール。

チュートリアルを完了するための前提条件については、「JDBC チュートリアルの前提条件」を参照してください。

JDBC チュートリアルの前提条件

「AWS Lake Formation JDBC チュートリアル」を開始する前に、以下を実行したことを確認してください。

- Lake Formation の使用の開始 の各タスクを完了する。
- チュートリアルで使用する、JDBC がアクセスできるデータストアを決定する。
- JDBC タイプの AWS Glue 接続を作成するために必要な情報を収集する。この Data Catalog オブジェクトには、データストアへの URL とログイン認証情報が含まれ、データストアが Amazon Virtual Private Cloud (Amazon VPC) で作成された場合は、追加の VPC 固有の設定情報も含まれます。詳細については、「AWS Glue デベロッパーガイド」の「AWS Glue データカタログでの接続の定義」を参照してください。

このチュートリアルでは、 AWS Identity and Access Management (IAM) に精通していることを前提としています。IAM については、「IAM ユーザーガイド」を参照してください。

開始するには、「the section called “ステップ 1: データアナリストユーザーの作成”」に進んでください。

ステップ 1: データアナリストユーザーの作成

このステップでは、データレイクのデータアナリストとなる AWS Identity and Access Management (IAM) ユーザーを作成します AWS Lake Formation。

このユーザーは、データレイクをクエリするための最小限の許可セットを持っています。

1. IAM コンソール (https://console.aws.amazon.com/iam) を開きます。`AdministratorAccess` AWS 管理ポリシーを使用してユーザーとしてサインインします。
2. 以下の設定で、`datalake_user` という名前のユーザーを作成します。
   - AWS マネジメントコンソール アクセスを有効にします。
   - パスワードを設定して、パスワードのリセットを不要にする。
   - `AmazonAthenaFullAccess` AWS 管理ポリシーをアタッチします。
   - 以下のインラインポリシーをアタッチする。ポリシーには `DatalakeUserBasic` という名前を付けます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess",
                "glue:GetTable",
                "glue:GetTables",
                "glue:SearchTables",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetPartitions",
                "lakeformation:GetResourceLFTags",
                "lakeformation:ListLFTags",
                "lakeformation:GetLFTag",
                "lakeformation:SearchTablesByLFTags",
                "lakeformation:SearchDatabasesByLFTags"
           ],
            "Resource": "*"
        }
    ]
}
```

ステップ 2: AWS Glue で接続を作成する

> **Note**
>
> JDBC データソースへの AWS Glue 接続がすでに作成されている場合は、このステップをスキップしてください。

AWS Lake Formation は、 AWS Glue接続を介して JDBC データソースにアクセスします。接続は、データソースへの接続に必要となるすべての情報が含まれた Data Catalog オブジェクトです。接続は、AWS Glue コンソールを使用して作成することができます。

接続を作成する

1. でAWS Glueコンソールを開きhttps://console.aws.amazon.com/glue/、管理者権限を持つユーザーとしてサインインします。
2. ナビゲーションペインの [データカタログ] で [接続] を選択します。
3. コネクタ ページで、[カスタムコネクタを作成] を選択します。
4. [データソースの選択] ページで、接続タイプとして [JDBC] を選択します。その後、[Next] (次へ) を選択します。
5. 接続ウィザードを続けて実行し、接続を保存します。

   接続の作成に関する詳細については、「AWS Glue デベロッパーガイド」の「AWS Glue JDBC 接続プロパティ」を参照してください。

ステップ 3: データレイク用の Amazon S3 バケットを作成する

このステップでは、データレイクのルートロケーションになる Amazon Simple Storage Service (Amazon S3) バケットを作成します。

1. https://console.aws.amazon.com/s3/ で Amazon S3 コンソールを開き、管理者権限を持つユーザーとしてサインインします。
2. [Create bucket] (バケットを作成) を選択し、ウィザードをすべて実行して *<yourName>*-datalake-tutorial という名前のバケットを作成します。*<yourName>* はユーザーの名前のイニシャルと苗字の組み合わせです。例: jdoe-datalake-tutorial。

Amazon S3 バケットの作成に関する詳しい手順については、「Amazon Simple Storage Service ユーザーガイド」の「S3 バケットの作成方法」を参照してください。

ステップ 4: Amazon S3 パスを登録する

このステップでは、Amazon Simple Storage Service (Amazon S3) パスをデータレイクのルートロケーションとして登録します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。データレイク管理者としてサインインします。
2. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。
3. [Register location] (ロケーションを登録) を選択してから、[Browse] (参照) を選択します。
4. 前に作成した *<yourName>*-datalake-tutorial バケットを選択し、デフォルトの IAM ロール AWSServiceRoleForLakeFormationDataAccess を受け入れ、[Register location] (ロケーションを登録) を選択します。

   ロケーションの登録に関する詳細については、「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

ステップ 5: データのロケーションに対する許可を付与する

プリンシパルは、作成する Data Catalog のテーブルまたはデータベースのポイント先となるデータレイクロケーションに対するデータロケーション許可を持っている必要があります。ワークフローの IAM ロールにデータロケーション許可を付与して、ワークフローがデータ取り込み先に書き込みを実行できるようにする必要があります。

1. Lake Formation コンソールのナビゲーションペインにある [Permissions] (許可) で [Data locations] (データのロケーション) を選択します。
2. [Grant] (付与) を選択し、[Grant permissions] (許可の付与) ダイアログボックスで以下を実行します。

   a. [IAM user and roles] (IAM ユーザーおよびロール) で、LakeFormationWorkflowRole を選択します。

   b. [Storage locations] (ストレージのロケーション) で、使用する *<yourName>*-datalake-tutorial バケットを選択します。

3. [Grant] (付与) を選択します。

データロケーション許可については、「Underlying data access control」を参照してください。

ステップ 6: Data Catalog でデータベースを作成する

Lake Formation Data Catalog のメタデータテーブルは、データベース内に保存されます。

1. Lake Formation コンソールのナビゲーションペインにある [Data catalog] で [Databases] (データベース) を選択します。
2. [Create database] (データベースを作成) を選択し、[Database details] (データベースの詳細) で `lakeformation_tutorial` という名前を入力します。
3. 他のフィールドは空欄のままにしておき、[Create database] (データベースを作成) を選択します。

ステップ 7: データの許可を付与する

Data Catalog でメタデータテーブルを作成するための許可を付与する必要があります。ワークフローは `LakeFormationWorkflowRole` ロールを使用して実行されるため、これらの許可をロールに付与する必要があります。

1. Lake Formation コンソールのナビゲーションペインにある [許可] で [データレイクのアクセス許可] を選択します。
2. [Grant] (付与) を選択し、[Grant data permissions] (データ許可の付与) ダイアログボックスで以下を実行します。
   a. [Principals] (プリンシパル) の [IAM user and roles] (IAM ユーザーおよびロール) で `LakeFormationWorkflowRole` を選択します。
   b. [LF-Tags or catalog resources] (LF タグまたはカタログリソース) で、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。
   c. [Databases] (データベース) には、前に作成したデータベースである `lakeformation_tutorial` を選択します。
   d. [データベースのアクセス許可] で、[テーブルを作成]、[変更]、および [ドロップ] をオンにして、[スーパー] が選択されている場合はそれをオフにします。
3. [Grant] (付与) を選択します。

Lake Formation 許可の付与に関する詳細ついては、「Lake Formation 許可の概要」を参照してください。

ステップ 8: ブループリントを使用してワークフローを作成する

AWS Lake Formation ワークフローは、データレイクを検出して取り込むAWS Glueジョブ、クローラ、トリガーを生成します。ワークフローは、事前定義された Lake Formation ブループリントのいずれかに基づいて作成します。

1. Lake Formation コンソールのナビゲーションペインで [Blueprints] (ブループリント) を選択してから、[Use blueprint] (ブループリントを使用) を選択します。
2. [Use a blueprint] (ブループリントの使用) ページにある [Blueprint type] (ブループリントタイプ) で [Database snapshot] (データベーススナップショット) を選択します。
3. [Import source] (インポートソース) の [Database connection] (データベース接続) には、先ほど作成した接続である `datalake-tutorial`、またはデータソースの既存の接続を選択します。
4. [Source data path] (ソースデータパス) には、データの取り込み元となるパスを *<database>*/*<schema>*/*<table>* の形式で入力します。

   スキーマまたはテーブルの代わりに、パーセント (%) ワイルドカードを使用することができます。スキーマをサポートするデータベースの場合は、*<database>* 内の *<schema>* にあるすべてのテーブルと一致させるために、*<database>*/*<schema>*/% を入力します。Oracle データベースと MySQL はパス内のスキーマをサポートしないので、代わりに *<database>*/% を入力します。Oracle データベースの場合、*<database>* はシステム識別子 (SID) です。

   例えば、Oracle データベースの SID が `orcl` の場合は、`orcl/%` を入力して、JDCB 接続で指定されたユーザーがアクセスできるすべてのテーブルと一致させます。

   > **Important**
   >
   > このフィールドでは、大文字と小文字が区別されます。

5. [Import target] (インポートターゲット) で、以下のパラメータを指定します。

[Target database] (ターゲットデータベース)	`lakeformation_tutorial`
[Target storage location] (ターゲットストレージロケーション)	`s3://`*<yourName>* `-datalake-tutorial`

[Data format] (データ形式)	(Parquet または CSV を選択)

6. [Import Frequency] (インポート頻度) には、[Run on demand] (オンデマンドで実行) を選択します。
7. [Import target] (インポートオプション) で、以下のパラメータを指定します。

[Workflow name] (ワークフロー名)	`lakeformationjdbctest`
[IAM role] (IAM ロール)	`LakeFormationWorkflowRole`
[Table prefix] (テーブルプレフィックス)	`jdbctest` **Note** 小文字を使用する必要があります。

8. [Create] (作成) を選択し、ワークフローが正常に作成されたことコンソールが報告するまで待機します。

Tip

以下のエラーメッセージが表示されましたか?
`User: arn:aws:iam::<account-id>:user/<datalake_administrator_user> is not authorized to perform: iam:PassRole on resource:arn:aws:iam::<account-id>:role/LakeFormationWorkflowRole...`
その場合、データレイク管理者ユーザーのインラインポリシーで *<account-id>* を有効な AWS アカウント番号に置き換えたことを確認します。

ステップ 9: ワークフローを実行する

ワークフローをrun-on-demandするように指定したため、ワークフローを手動で開始する必要があります AWS Lake Formation。

1. Lake Formation コンソールの [Blueprints] (ブループリント) ページで、ワークフロー `lakeformationjdbctest` を選択します。

2. [Actions] (アクション) を選択してから、[Start] (開始) を選択します。
3. ワークフローの実行に伴って、その進捗状況を [Last run status] (最終実行ステータス) 列で確認します。更新ボタンを随時選択します。

   ステータスは、[RUNNING] (実行中) から、[Discovering] (検出中)、[Importing] (インポート中)、[COMPLETED] (完了) と移行します。

   ワークフローが完了すると、以下のようになります。

   - Data Catalog に新しいメタデータテーブルがある。
   - データがデータレイクに取り込まれる。

   ワークフローが失敗する場合は、以下を実行します。

   a. ワークフローを選択します。[Actions] (アクション) を選択してから、[View graph] (グラフを表示) を選択します。

      AWS Glue コンソールでワークフローが開きます。
   b. ワークフローを選択し、[History] (履歴) タブを選択します。
   c. 最新の実行を選択し、[View run details] (実行の詳細を表示) を選択します。
   d. 動的 (ランタイム) グラフで失敗したジョブまたはクローラを選択し、エラーメッセージを確認します。障害が発生したノードは赤色または黄色のいずれかになっています。

ステップ 10: テーブルに対する SELECT を付与する

データアナリストがテーブルが指すデータをクエリ AWS Lake Formation できるように、 の新しいデータカタログテーブルに対する アクセスSELECT許可を付与する必要があります。

> **Note**
>
> ワークフローは、ワークフローが作成するテーブルに対する SELECT 許可を、ワークフローを実行したユーザーに自動的に付与します。このワークフローはデータレイク管理者が実行したので、データアナリストに SELECT を付与する必要があります。

1. Lake Formation コンソールのナビゲーションペインにある [許可] で [データレイクのアクセス許可] を選択します。

2. [Grant] (付与) を選択し、[Grant data permissions] (データ許可の付与) ダイアログボックスで以下を実行します。

   a. [Principals] (プリンシパル) の [IAM user and roles] (IAM ユーザーおよびロール) で `datalake_user` を選択します。

   b. [LF-Tags or catalog resources] (LF タグまたはカタログリソース) で、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。

   c. [Database] (データベース) には `lakeformation_tutorial` を選択します。

      [Tables] (テーブル) リストが表示されます。

   d. [Tables] (テーブル) には、データソースから 1 つ、または複数のテーブルを選択します。

   e. [Table and column permissions] (テーブルと列の許可) で [Select] (選択) をオンにします。

3. [Grant] (付与) を選択します。

次のステップは、データアナリストとして実行します。

ステップ 11: を使用してデータレイクをクエリする Amazon Athena

Amazon Athena コンソールを使用して、データレイク内のデータをクエリします。

1. Athena コンソール (https://console.aws.amazon.com/athena/) を開き、データアナリストであるユーザー `datalake_user` としてサインインします。
2. 必要に応じて [Get Started] (開始する) を選択して、Athena クエリエディタに進みます。
3. [Data source] (データソース) で [AwsDataCatalog] を選択します。
4. [Database] (データベース) で、`lakeformation_tutorial` を選択します。

   [Tables] (テーブル) リストが表示されます。

5. テーブルの 1 つの横にあるポップアップメニューで、[Preview table] (テーブルのプレビュー) を選択します。

   クエリが実行され、10 行のデータが表示されます。

ステップ 12: Amazon Redshift Spectrum を使用してデータレイク内のデータをクエリする

Amazon Simple Storage Service (Amazon S3) データレイクにインポートしたデータをクエリするように Amazon Redshift Spectrum をセットアップすることができます。まず、Amazon Redshift クラスターを起動し、Amazon S3 データをクエリするために使用される AWS Identity and Access Management (IAM) ロールを作成します。 Amazon S3 次に、このロールにクエリを実行するテーブルに対する `Select` 許可を付与します。その後、Amazon Redshift クエリエディタを使用する許可をユーザーに付与します。最後に、Amazon Redshift クラスターを作成して、クエリを実行します。

管理者としてクラスターを作成し、データアナリストとしてクラスターをクエリします。

Amazon Redshift Spectrum の詳細については、「Amazon Redshift データベースデベロッパーガイド」の「Amazon Redshift Spectrum を使用した外部データのクエリ」を参照してください。

Amazon Redshift クエリを実行する許可をセットアップする

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。`AdministratorAccess` AWS 管理ポリシーを使用してユーザーとしてサインインします。
2. ナビゲーションペインで [Policies] (ポリシー) を選択します。

   [Policies] (ポリシー) を初めて選択する場合は、[Welcome to Managed Policies] (マネージドポリシーにようこそ) ページが表示されます。[Get Started] (今すぐ始める) を選択します。
3. [Create policy] (ポリシーを作成) を選択します。
4. [JSON] タブを選択します。
5. 以下の JSON ポリシードキュメントを貼り付けます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess",
                "glue:GetTable",
                "glue:GetTables",
                "glue:SearchTables",
                "glue:GetDatabase",
                "glue:GetDatabases",
```

```
                "glue:GetPartitions",
                "lakeformation:GetResourceLFTags",
                "lakeformation:ListLFTags",
                "lakeformation:GetLFTag",
                "lakeformation:SearchTablesByLFTags",
                "lakeformation:SearchDatabasesByLFTags"
            ],
             "Resource": "*"
        }
    ]
}
```

6. 完了したら、[Review] (確認) を選択してポリシーを確認します。構文エラーがある場合は、ポリシーバリデータが報告します。
7. [Review policy] (ポリシーの確認) ページで、作成しているポリシーの [Name] (名前) に **`RedshiftLakeFormationPolicy`** を入力します。[Description] (説明) を入力します (オプション)。ポリシーの [Summary] (概要) を参照して、ポリシーによって付与された許可を確認します。次に、[Create policy] (ポリシーの作成) を選択して作業を保存します。
8. IAM コンソールのナビゲーションペインで、[Roles] (ロール)、[Create role] (ロールを作成) の順に選択します。
9. [Select trusted entity] (信頼されたエンティティの選択) で、[AWS のサービス] を選択します。
10. [Amazon Redshift] サービスを選択して、このロールを引き受けます。
11. サービスのユースケースに [Redshift - Customizable] (Redshift – カスタマイズ可能) を選択します。その後、[Next] (次へ) を選択します。
12. 作成した許可ポリシーである `RedshiftLakeFormationPolicy` を検索して、リスト内のそのポリシー名の横にあるチェックボックスをオンにします。
13. [Next: Tags] (次のステップ: タグ) を選択します。
14. [Next: Review] (次のステップ: レビュー) を選択します。
15. [Role name] (ロール名) に名前 **`RedshiftLakeFormationRole`** を入力します。
16. (オプション) [Role description] (ロールの説明) に、新しいロールの説明を入力します。
17. ロールを確認してから、[Create role] (ロールを作成) を選択します。

Lake Formation データベース内でクエリされるテーブルに対する **`Select`** 許可を付与します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。データレイク管理者としてサインインします。

2. ナビゲーションペインの [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択して、[Grant] (付与) を選択します。
3. 以下の情報を指定します。

   - [IAM users and roles] (IAM ユーザーおよびロール) には、作成した IAM ロールである `RedshiftLakeFormationRole` を選択します。Amazon Redshift クエリエディタを実行するときは、データに対する許可にこの IAM ロールが使用されます。
   - [Database] (データベース) で、`lakeformation_tutorial` を選択します。

     テーブルのリストが表示されます。
   - [Table] (テーブル) には、クエリするデータソース内のテーブルを選択します。
   - [Select] (選択) テーブル許可をオンにします。
4. [Grant] (付与) を選択します。

Amazon Redshift Spectrum をセットアップしてクエリを実行する

1. Amazon Redshift コンソール (https://console.aws.amazon.com/redshift) を開きます。ユーザー `Administrator` としてサインインします。
2. [Create cluster] (クラスターを作成) を選択します。
3. [Create cluster] (クラスターを作成) ページで、[Cluster identifier] (クラスター識別子) に `redshift-lakeformation-demo` を入力します。
4. [Node type] (ノードの種類) には、[dc2.large] を選択します。
5. スクロールダウンして、[Database configurations] (データベース設定) で、これらのパラメータを入力、または受け入れます。

   - [Admin user name] (管理者ユーザー名): `awsuser`
   - [Admin user password] (管理者ユーザーパスワード): (*`Choose a password`*)
6. [Cluster permissions] (クラスターの許可) を展開し、[Available IAM roles] (利用可能な IAM ロール) で [RedshiftLakeFormationRole] を選択します。次に、[Add IAM role] (IAM ロールを追加) を選択します。
7. デフォルト値である 5439 とは異なるポートを使用する必要がある場合は、[Additional configurations] (追加設定) の横にある [Use defaults] (デフォルトを使用) オプションをオフにします。[Database configurations] (データベース設定) のセクションを展開し、新しい [Database port] (データベースポート) 番号を入力します。
8. [Create cluster] (クラスターを作成) を選択します。

[Clusters] (クラスター) ページがロードされます。

9. クラスターのステータスが [Available] (利用可能) になるまで待ちます。更新アイコンを定期的に選択します。
10. クラスターに対してクエリを実行する許可をデータアナリストに付与します。これには、以下のステップを実行します。

    a. IAM コンソール (https://console.aws.amazon.com/iam/) を開き、`Administrator` ユーザーとしてサインインします。

    b. ナビゲーションペインで [Users] (ユーザー) を選択し、ユーザー `datalake_user` に以下のマネージドポリシーをアタッチします。

    - `AmazonRedshiftQueryEditor`
    - `AmazonRedshiftReadOnlyAccess`

11. Amazon Redshift のコンソールからサインアウトし、ユーザー `datalake_user` として再度サインインします。
12. 左にある垂直ツールバーで [Query Editor] (クエリエディタ) アイコンを選択してクエリエディタを開き、クラスターに接続します。[Connect to database] (データベースに接続) ダイアログボックスが表示されたら、クラスター名 `redshift-lakeformation-demo` を選択し、作成したデータベース名 **`dev`**、ユーザー名 **`awsuser`**、およびパスワードを入力します。[Connect to database] (データベースに接続) を選択します。

    > **Note**
    >
    > 接続パラメータのプロンプトが表示されず、クエリエディタで別のクラスターがすでに選択されている場合は、[Change Connection] (接続を変更) を選択して、[Connect to database] (データベースに接続) ダイアログボックスを開きます。

13. 新しい [Query 1] (クエリ 1) テキストボックスに以下のステートメントを入力して実行し、Lake Formation のデータベース `lakeformation_tutorial` を Amazon Redshift スキーマ名 `redshift_jdbc` にマップします。

    > **Important**
    >
    > *`<account-id>`* を有効な AWS アカウント番号に、*`<region>`* を有効な AWS リージョン名 (例:) に置き換えますus-east-1。

```
create external schema if not exists redshift_jdbc from DATA CATALOG
 database 'lakeformation_tutorial' iam_role 'arn:aws:iam::<account-id>:role/
RedshiftLakeFormationRole' region '<region>';
```

14. [Select schema] (スキーマの選択) にあるスキーマリストで、[redshift_jdbc] を選択します。

    テーブルのリストが表示されます。クエリエディタには、Lake Formation データレイク許可が付与されたテーブルのみが表示されます。

15. テーブル名の横にあるポップアップメニューで、[Preview data] (データをプレビュー) を選択します。

    Amazon Redshift は最初の 10 行を返します。

    これで、許可を持っているテーブルと列に対してクエリを実行できるようになりました。

ステップ 13: Amazon Redshift Spectrum を使用して Lake Formation 許可を付与または取り消す

Amazon Redshift は、変更された SQL ステートメントを使用してデータベースとテーブルに対する Lake Formation 許可の付与と取り消しを実行する機能をサポートします。これらのステートメントは、既存の Amazon Redshift ステートメントに似ています。詳細については、「Amazon Redshift データベースデベロッパーガイド」の「GRANT」と「REVOKE」を参照してください。

Lake Formation でのオープンテーブルストレージフォーマットのアクセス許可の設定

AWS Lake Formation は、Apache Iceberg、Apache Hudi、Linux 基盤 Delta Lake などの Open Table Formats (OTFs) のアクセス許可の管理をサポートしています。このチュートリアルでは、 AWS Glue Data Catalog でシンボリックリンクマニフェストテーブルを使用して Iceberg、Hudi、Delta Lake を作成し AWS Glue、Lake Formation を使用してきめ細かなアクセス許可を設定し、Amazon Athena を使用してデータをクエリする方法について説明します。

> **Note**
>
> AWS 分析サービスは、すべてのトランザクションテーブル形式をサポートしているわけではありません。詳細については、「他の AWS サービスの使用」を参照してください。この

チュートリアルでは、 AWS Glue ジョブのみを使用して、データカタログに新しいデータベースとテーブルを手動で作成する方法について説明します。

このチュートリアルには、クイックセットアップ用の AWS CloudFormation テンプレートが含まれています。このテンプレートを参照し、ニーズに合わせてカスタマイズできます。

トピック

- 対象者
- 前提条件
- ステップ 1: リソースをプロビジョニングする
- ステップ 2: Iceberg テーブルのアクセス許可をセットアップする
- ステップ 3: Hudi テーブルのアクセス許可をセットアップする
- ステップ 4: Delta Lake テーブルのアクセス許可をセットアップする
- ステップ 5: AWS リソースをクリーンアップする

対象者

このチュートリアルは、IAM 管理者、データレイク管理者、ビジネスアナリストを対象としています。次の表は、このチュートリアルで Lake Formation による管理対象テーブルの作成に使用するロールのリストです。

ロール	説明
IAM 管理者	IAM ユーザーおよびロール、Amazon S3 バケットを作成できるユーザー。`AdministratorAccess` AWS 管理ポリシーがあります。
データレイク管理者	Data Catalog へのアクセス、データベースの作成、および他のユーザーへの Lake Formation 許可の付与を実行できるユーザー。IAM 許可の数は IAM 管理者よりも少ないですが、データレイクを管理するには十分な許可を持っています。

ロール	説明
ビジネスアナリスト	データレイクに対してクエリを実行できるユーザー。クエリを実行するためのアクセス許可を持っています。

前提条件

このチュートリアルを開始する前に、正しいアクセス許可を持つユーザーとしてサインイン AWS アカウント できる が必要です。詳細については、「にサインアップする AWS アカウント」を参照してください。

このチュートリアルでは、ユーザーが IAM のロールおよびポリシーに精通していることを前提としています。IAM については、「IAM ユーザーガイド」を参照してください。

このチュートリアルを完了するには、次の AWS リソースを設定する必要があります。

- データレイク管理ユーザー
- Lake Formation データレイクの設定
- Amazon Athena エンジンバージョン 3

データレイク管理者を作成するには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) に管理者ユーザーとしてサインインします。このチュートリアルでは、米国東部 (バージニア北部) リージョンにリソースを作成します。
2. ナビゲーションペインの Lake Formation コンソールの [許可] で [管理ロールとタスク] を選択します。
3. [データレイク管理者] で [管理者を選択] を選択します。
4. ポップアップウィンドウの [データレイク管理者の管理] の [IAM ユーザーとロール] で、[IAM 管理者ユーザー] を選択します。
5. [保存] を選択します。

データレイク設定を有効にするには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。ナビゲーションペインの [Data catalog] で [Settings] (設定) を選択します。次のチェックを外します。
   - 新しいデータベースには IAM アクセスコントロールのみを使用する。
   - 新しいデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する。
2. [クロスアカウントバージョン設定] で、クロスアカウントバージョンとして [バージョン 3] を選択します。
3. [保存] を選択します。

Amazon Athena エンジンをバージョン 3 にアップグレードするには

1. https://console.aws.amazon.com/athena/ で Athena コンソールを開きます。
2. [ワークグループ] を選択し、プライマリワークグループを選択します。
3. ワークグループのバージョンが 3 以上であることを確認してください。そうでない場合は、ワークグループを編集し、[クエリエンジンのアップグレード] で [手動] を選択し、バージョン 3 を選択します。
4. [Save changes] (変更の保存) をクリックします。

ステップ 1: リソースをプロビジョニングする

このセクションでは、 CloudFormation テンプレートを使用して AWS リソースを設定する方法について説明します。

CloudFormation テンプレートを使用してリソースを作成するには

1. 米国東部 (バージニア北部) リージョンの IAM 管理者として https://console.aws.amazon.com/cloudformation で AWS CloudFormation コンソールにサインインします。
2. [スタックの起動] を選択します。
3. [Create Stack] (スタックの作成) 画面で、[Next] (次へ) を選択します。
4. [Stack name] (スタック名) を入力します。
5. [次へ] を選択します。
6. 次のページで、[Next] (次へ) を選択します。

7. 最終ページの詳細を確認し、IAM リソースを作成する AWS CloudFormation 可能性があることを確認します。
8. [作成] を選択します。

   スタックの作成には、最大 2 分かかる場合があります。

クラウドフォーメーションスタックを起動すると、以下のリソースが作成されます。

- lf-otf-datalake-123456789012 – データを保存する Amazon S3 バケット

> **Note**
>
> Amazon S3 バケット名に追加されたアカウント ID は、アカウント ID に置き換えられます。

- lf-otf-tutorial-123456789012 – クエリ結果と AWS Glue ジョブスクリプトを保存する Amazon S3 バケット
- lficebergdb – AWS Glue Iceberg データベース
- lfhudidb – AWS Glue Hudi データベース
- lfdeltadb – AWS Glue デルタデータベース
- native-iceberg-create – データカタログに Iceberg テーブルを作成する AWS Glue ジョブ
- native-hudi-create – データカタログに Hudi テーブルを作成する AWS Glue ジョブ
- native-delta-create – データカタログに Delta テーブルを作成する AWS Glue ジョブ
- LF-OTF-GlueServiceRole – ジョブを実行する AWS Glue ために渡す IAM ロール。このロールには、Data Catalog、Amazon S3 バケットなどのリソースにアクセスするために必要なポリシーがアタッチされています。
- LF-OTF-RegisterRole – Amazon S3 ロケーションを Lake Formation に登録するための IAM ロール。このロールには、`LF-Data-Lake-Storage-Policy` が関連付けられています。
- lf-consumer-analystuser – Athena を使用してデータをクエリする IAM ユーザー
- lf-consumer-analystuser-credentials – に保存されているデータアナリストユーザーのパスワード AWS Secrets Manager

スタックの作成が完了したら、出力タブに移動して、次の値を書き留めます。

- AthenaQueryResultLocation – Athena クエリ出力の Amazon S3 ロケーション

- BusinessAnalystUserCredentials – データアナリストユーザーのパスワード

  パスワード値を取得するには:

  1. Secrets Manager コンソールに移動して、`lf-consumer-analystuser-credentials` 値を選択します。
  2. [シークレット値] セクションで、[シークレット値の取得] を選択します。
  3. パスワードのシークレット値を書き留めておきます。

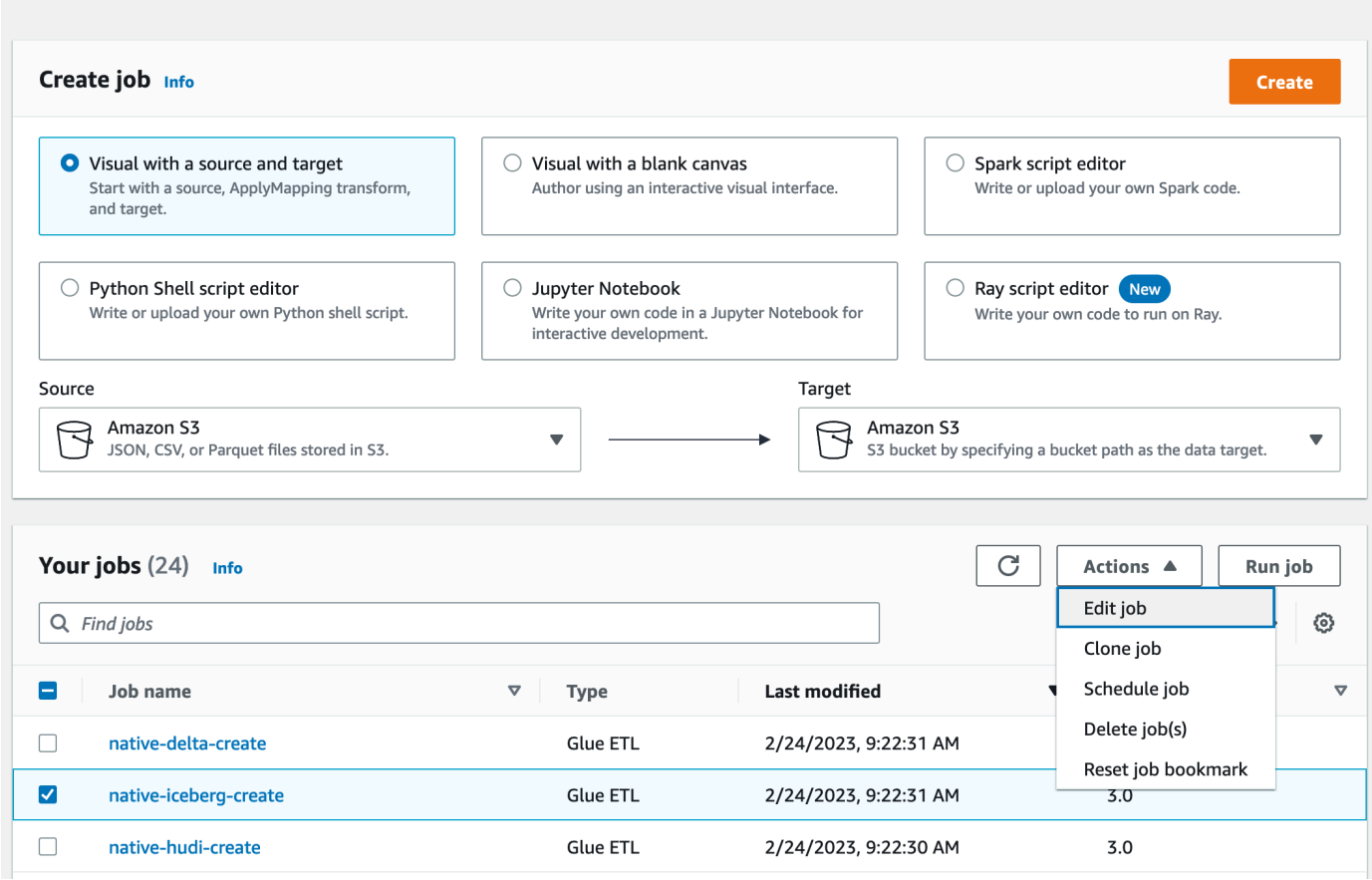
ステップ 2: Iceberg テーブルのアクセス許可をセットアップする

このセクションでは、 で Iceberg テーブルを作成し AWS Glue Data Catalog、 でデータアクセス許可を設定し AWS Lake Formation、Amazon Athena を使用してデータをクエリする方法について説明します。

Iceberg テーブルを作成するには

このステップでは、データカタログに Iceberg トランザクションテーブルを作成する AWS Glue ジョブを実行します。

1. データレイク管理者ユーザーとして、米国東部 (バージニア北部) リージョンの https://console.aws.amazon.com/glue/ で AWS Glue 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、[ジョブ] を選択します。
3. `native-iceberg-create` を選択します。



4. [アクション] で [ジョブの編集] を選択します。
5. ジョブの詳細で、高度なプロパティを展開し、Hive メタストア AWS Glue Data Catalog として使用 の横にあるチェックボックスをオンにして、 にテーブルメタデータを追加します AWS Glue Data Catalog。これは、ジョブで使用される Data Catalog リソースのメタストア AWS Glue Data Catalog として を指定し、後で Lake Formation のアクセス許可をカタログリソースに適用できるようにします。
6. [保存] を選択します。
7. [Run (実行)] を選択します。実行中、ジョブのステータスを表示できます。

   AWS Glue ジョブの詳細については、「 AWS Glue デベロッパーガイド」の「 AWS Glue コンソールでのジョブの使用」を参照してください。

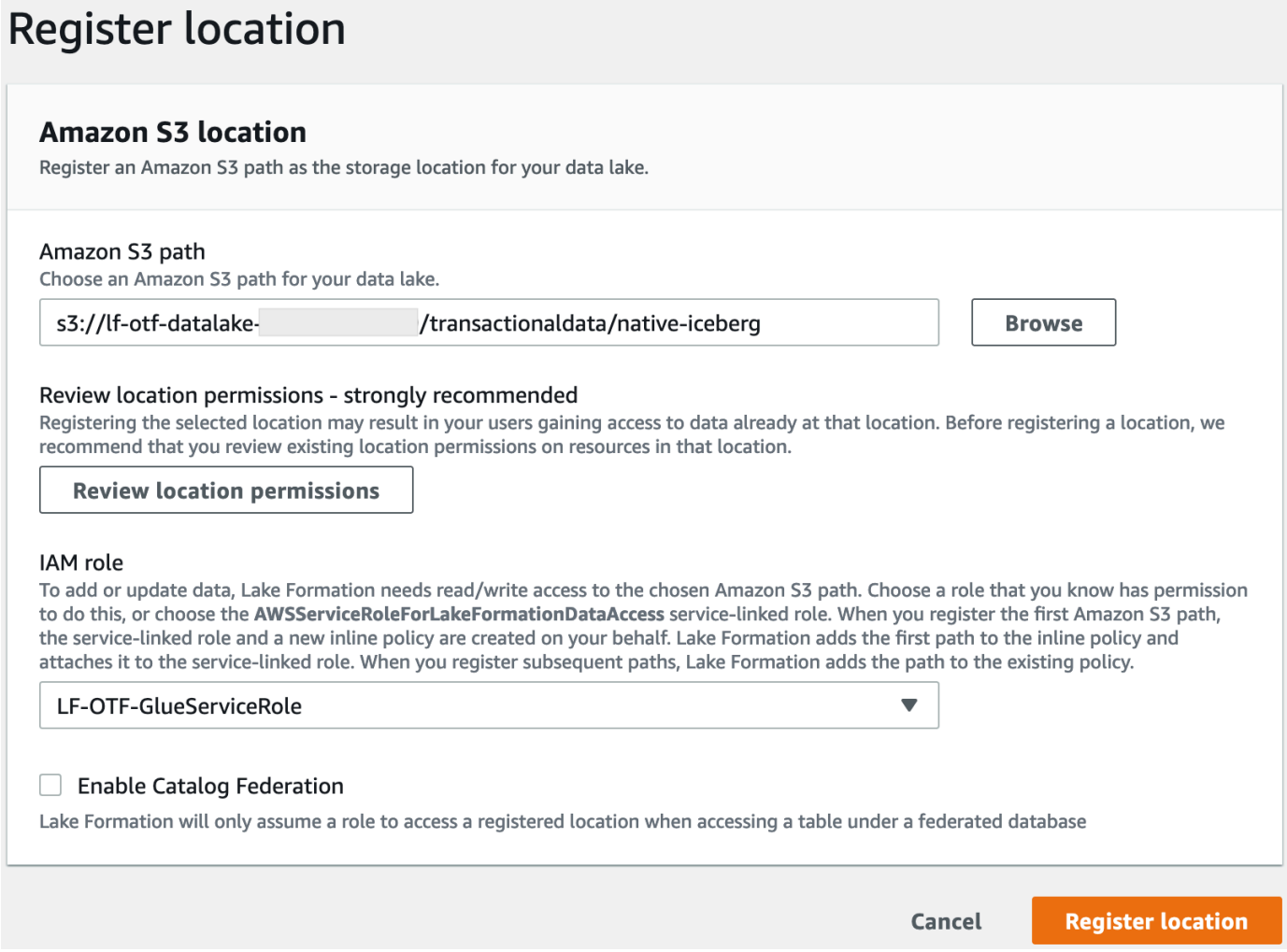
   このジョブは、`lficebergdb` データベースに `product` という名前を付けた Iceberg テーブルを作成します。Lake Formation コンソールの製品テーブルを確認してください。

データロケーションを Lake Formation に登録するには

次に、Amazon S3 パスをデータレイクのロケーションとして登録します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) をデータレイク管理者ユーザーとして開きます。
2. ナビゲーションペインの [登録および取り込み] で [データレイクのロケーション] を選択します。
3. [Register location] (ロケーションを登録) を選択します。
4. [ロケーションを登録] ページで、次のように入力します。

   - [Amazon S3 パス] – [ブラウズ] を選択して `lf-otf-datalake-123456789012` を選択します。Amazon S3 ルートの場所の横にある展開矢印を選択して、`s3/buckets/lf-otf-datalake-123456789012/transactionaldata/native-icebergその場所に移動します。`
   - [IAM ロール] – IAM ロールとして `LF-OTF-RegisterRole` を選択します。
   - [Register location] (ロケーションを登録) を選択します。

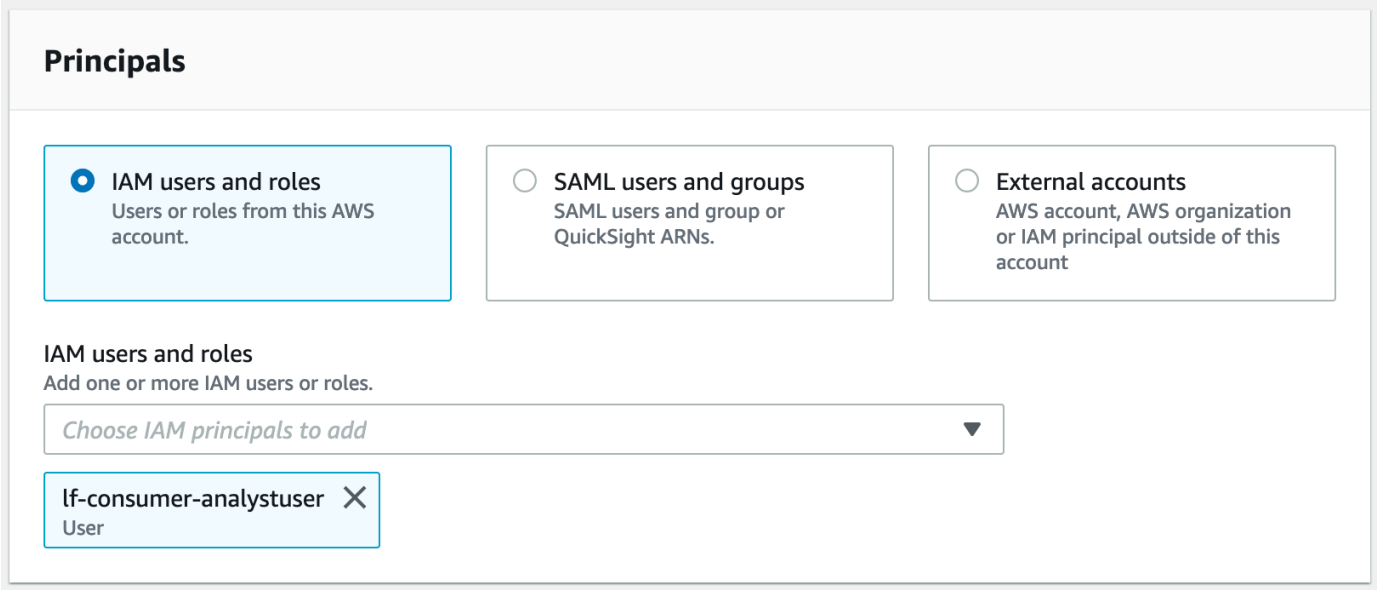


データロケーションを Lake Formation へ登録する方法の詳細については、「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

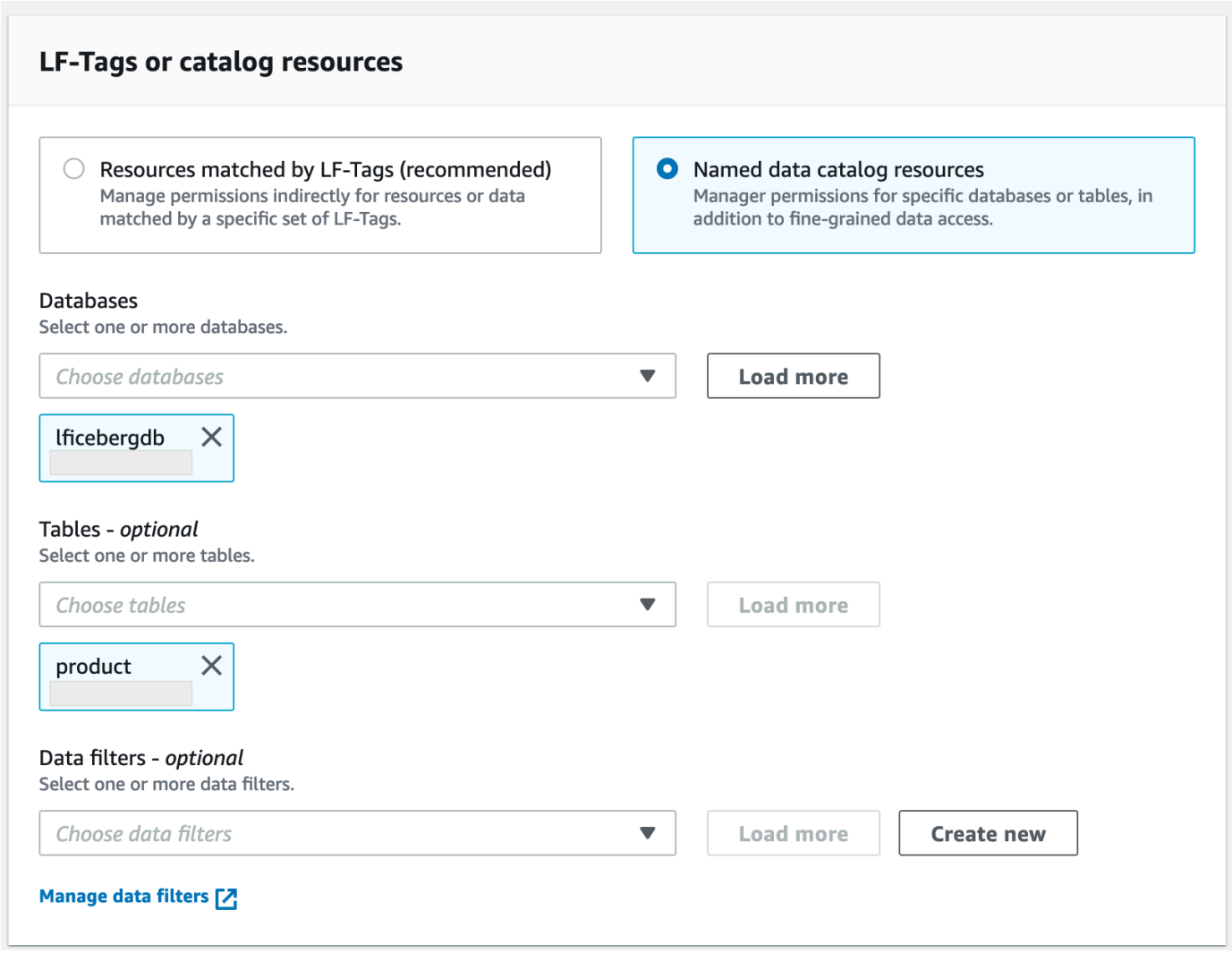
Iceberg テーブルで Lake Formation の権限を付与するには

このステップでは、ビジネスアナリストユーザーにデータレイクのアクセス許可を付与します。

1. [データレイクのアクセス許可] で、[付与] を選択します。
2. [データのアクセス許可の付与] 画面で、[IAM ユーザーとロール] を選択します。
3. ドロップダウンリストから [`lf-consumer-analystuser`] を選択します。



4. [名前付きの Data Catalog リソース] を選択します。
5. [データベース] には `lficebergdb` を選択します。
6. [Tables] (テーブル) には `product` を選択します。



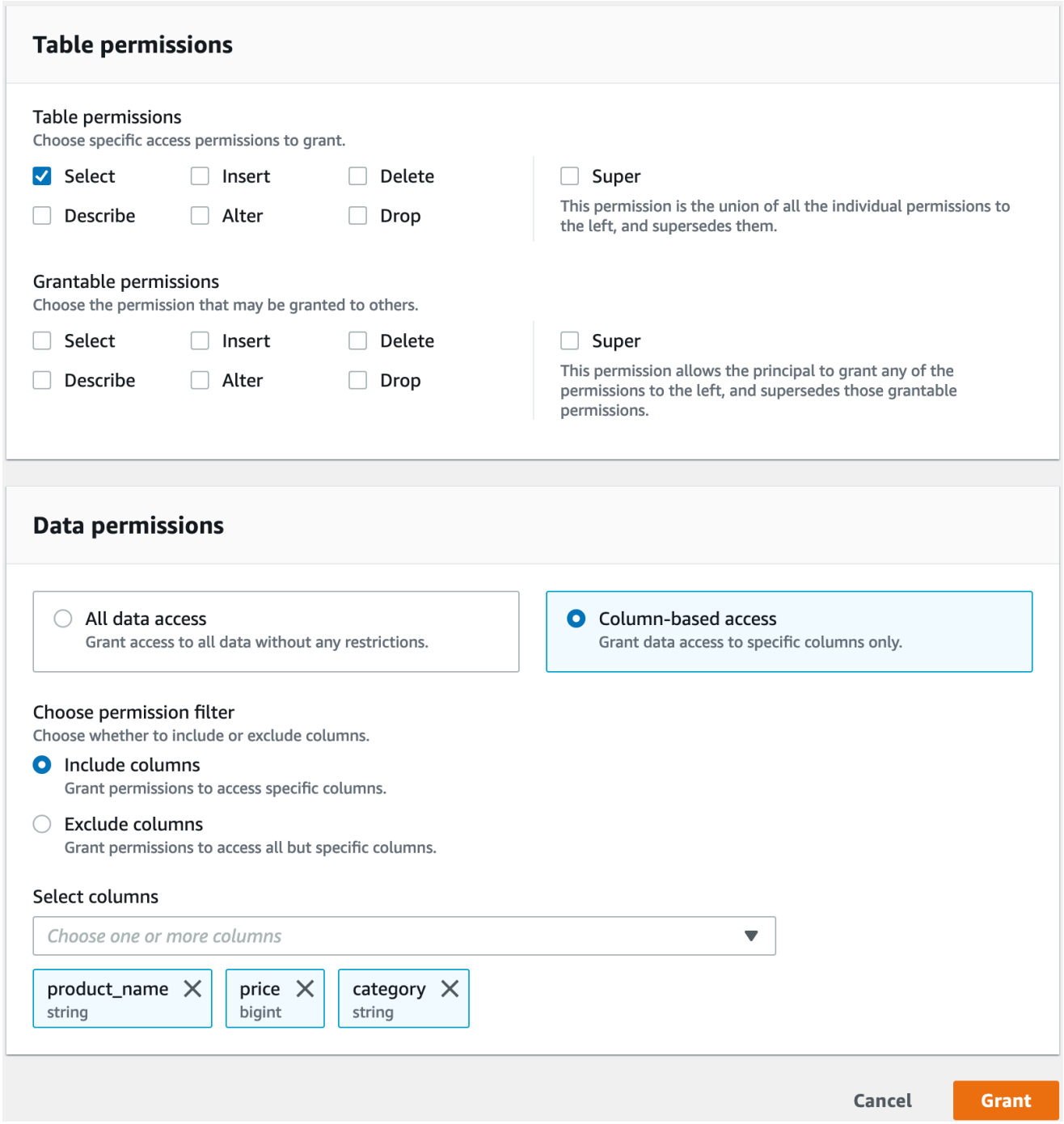
7. 次に、列を指定して列ベースのアクセスを許可できます。

   a. [テーブル許可] には [選択] を選択します。

   b. [データのアクセス許可] で [列ベースのアクセス] を選択し、[列を含める] を選択します。

   c. `product_name`、`price`、`category` 列を選択します。

   d. [付与] を選択します。



Athena を使用して Iceberg テーブルをクエリするには

ここで Athena を使用し、作成した Iceberg テーブルに対するクエリを開始します。初めて Athena でクエリを実行する場合は、クエリ結果の場所を設定する必要があります。詳細については、「クエリ結果の場所の指定」を参照してください。

1. データレイク管理者ユーザーとしてサインアウトし、 CloudFormation 出力から前に書き留めたパスワードを使用して、米国東部 (バージニア北部) リージョンlf-consumer-analystuserで としてサインインします。
2. https://console.aws.amazon.com/athena/ で Athena コンソールを開きます。
3. [設定] を選択し、[管理] を選択します。
4. クエリ結果の場所 ボックスに、 CloudFormation 出力で作成したバケットへのパスを入力します。AthenaQueryResultLocation (s3://lf-otf-tutorial-123456789012/athena-results/) の値をコピーして、[保存] を選択します。
5. 次のクエリを実行して、Iceberg テーブルに保存されている 10 個のレコードをプレビューします。

```
select * from lficebergdb.product limit 10;
```

Athena を使用して Iceberg テーブルをクエリする方法の詳細については、「Amazon Athena ユーザーガイド」の「Iceberg テーブルへのクエリ」を参照してください。

ステップ 3: Hudi テーブルのアクセス許可をセットアップする

このセクションでは、 で Hudi テーブルを作成し AWS Glue Data Catalog、 でデータアクセス許可を設定し AWS Lake Formation、Amazon Athena を使用してデータをクエリする方法について説明します。

Hudi テーブルを作成するには

このステップでは、データカタログに Hudi トランザクションテーブルを作成する AWS Glue ジョブを実行します。

1. 米国東部 (バージニア北部) リージョンで AWS Glue コンソール (https://console.aws.amazon.com/glue/) にサインインします。

   データレイク管理ユーザーとして開きます。
2. 左側のナビゲーションペインで、[ジョブ] を選択します。
3. native-hudi-create を選択します。
4. [アクション] で [ジョブの編集] を選択します。
5. ジョブの詳細で、高度なプロパティを展開し、Hive メタストア AWS Glue Data Catalog として使用 の横にあるチェックボックスをオンにして、 にテーブルメタデータを追加します AWS

Glue Data Catalog。これは、ジョブで使用される Data Catalog リソースのメタストア AWS Glue Data Catalog として を指定し、後で Lake Formation のアクセス許可をカタログリソースに適用できるようにします。

6. [保存] を選択します。
7. [Run (実行)] を選択します。実行中、ジョブのステータスを表示できます。

   AWS Glue ジョブの詳細については、「 AWS Glue デベロッパーガイド」の「 AWS Glue コンソールでのジョブの使用」を参照してください。

   このジョブは、データベース:lfhudidb に Hudi(cow) テーブルを作成します。Lake Formation コンソールの `product` テーブルを確認してください。

データロケーションを Lake Formation に登録するには

次に、Amazon S3 パスをデータレイクのルートロケーションとして登録します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にデータレイク管理者ユーザーとしてサインインします。
2. ナビゲーションペインの [登録および取り込み] で [データレイクのロケーション] を選択します。
3. [Register location] (ロケーションを登録) を選択します。
4. [ロケーションを登録] ページで、次のように入力します。

   - [Amazon S3 パス] – [ブラウズ] を選択して `lf-otf-datalake-123456789012` を選択します。Amazon S3 ルートの場所の横にある展開矢印を選択して、`s3/buckets/lf-otf-datalake-123456789012/transactionaldata/native-hudi`その場所に移動します。
   - [IAM ロール] – IAM ロールとして `LF-OTF-RegisterRole` を選択します。
   - [Register location] (ロケーションを登録) を選択します。

Hudi テーブルでデータレイクのアクセス許可を付与するには

このステップでは、ビジネスアナリストユーザーにデータレイクのアクセス許可を付与します。

1. [データレイクのアクセス許可] で、[付与] を選択します。
2. [データのアクセス許可の付与] 画面で、[IAM ユーザーとロール] を選択します。
3. ドロップダウンから `lf-consumer-analystuser`。

4. [名前付きのデータカタログリソース] を選択します。
5. [データベース] には `lfhudidb` を選択します。
6. [Tables] (テーブル) には `product` を選択します。
7. 次に、列を指定して列ベースのアクセスを許可できます。

   a. [テーブル許可] には [選択] を選択します。

   b. [データのアクセス許可] で [列ベースのアクセス] を選択し、[列を含める] を選択します。

   c. `product_name`、`price`、`category` 列を選択します。

   d. [付与] を選択します。

Athena を使用して Hudi テーブルをクエリするには

ここで Athena を使用し、作成した Hudi テーブルに対するクエリを開始します。初めて Athena でクエリを実行する場合は、クエリ結果の場所を設定する必要があります。詳細については、「クエリ結果の場所の指定」を参照してください。

1. データレイク管理者ユーザーとしてサインアウトし、 CloudFormation 出力から前に書き留めたパスワードを使用して、米国東部 (バージニア北部) リージョン`lf-consumer-analystuser`で としてサインインします。
2. https://console.aws.amazon.com/athena/ で Athena コンソールを開きます。
3. [設定] を選択し、[管理] を選択します。
4. クエリ結果の場所 ボックスに、 CloudFormation 出力で作成したバケットへのパスを入力します。`AthenaQueryResultLocation` (s3://lf-otf-tutorial-123456789012/athena-results/) の値をコピーして、[保存] します。
5. 次のクエリを実行して、Hudi テーブルに保存されている 10 個のレコードをプレビューします。

   ```
   select * from lfhudidb.product limit 10;
   ```

   Hudi テーブルをクエリする方法の詳細については、「Amazon Athena ユーザーガイド」の「Hudi テーブルのクエリ」を参照してください。

ステップ 4: Delta Lake テーブルのアクセス許可をセットアップする

このセクションでは、 でシンボリックリンクマニフェストファイルを使用して Delta Lake テーブルを作成し、 でデータアクセス許可を設定し AWS Glue Data Catalog、Amazon Athena を使用してデータを AWS Lake Formation クエリする方法について説明します。

Delta Lake テーブルを作成するには

このステップでは、データカタログに Delta Lake トランザクションテーブルを作成する AWS Glue ジョブを実行します。

1. 米国東部 (バージニア北部) リージョンで AWS Glue コンソール (https://console.aws.amazon.com/glue/) にサインインします。

   データレイク管理ユーザーとして開きます。
2. 左側のナビゲーションペインで、[ジョブ] を選択します。
3. `native-delta-create` を選択します。
4. [アクション] で [ジョブの編集] を選択します。
5. ジョブの詳細で、高度なプロパティを展開し、Hive メタストア AWS Glue Data Catalog として使用 の横にあるチェックボックスをオンにして、 にテーブルメタデータを追加します AWS Glue Data Catalog。これは、ジョブで使用される Data Catalog リソースのメタストア AWS Glue Data Catalog として を指定し、後で Lake Formation のアクセス許可をカタログリソースに適用できるようにします。
6. [保存] を選択します。
7. [アクション] で [実行] を選択します。

   このジョブは、`lfdeltadb` データベースに `product` という名前を付けた Delta Lake テーブルを作成します。Lake Formation コンソールの `product` テーブルを確認してください。

データロケーションを Lake Formation に登録するには

次に、Amazon S3 パスをデータレイクのルートロケーションとして登録します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) をデータレイク管理者ユーザーとして開きます。
2. ナビゲーションペインの [登録および取り込み] で [データレイクのロケーション] を選択します。

3. [Register location] (ロケーションを登録) を選択します。
4. [ロケーションを登録] ページで、次のように入力します。

   - [Amazon S3 パス] – [ブラウズ] を選択して `lf-otf-datalake-123456789012` を選択します。Amazon S3 ルートの場所の横にある展開矢印を選択して、`s3/buckets/lf-otf-datalake-123456789012/transactionaldata/native-delta`その場所に移動します。
   - [IAM ロール] – IAM ロールとして `LF-OTF-RegisterRole` を選択します。
   - [Register location] (ロケーションを登録) を選択します。

Delta Lake テーブルでデータレイクのアクセス許可を付与するには

このステップでは、ビジネスアナリストユーザーにデータレイクのアクセス許可を付与します。

1. [データレイクのアクセス許可] で、[付与] を選択します。
2. [データのアクセス許可の付与] 画面で、[IAM ユーザーとロール] を選択します。
3. ドロップダウンから `lf-consumer-analystuser`。
4. [名前付きのデータカタログリソース] を選択します。
5. [データベース] には `lfdeltadb` を選択します。
6. [Tables] (テーブル) には `product` を選択します。
7. 次に、列を指定して列ベースのアクセスを許可できます。

   a. [テーブル許可] には [選択] を選択します。
   b. [データのアクセス許可] で [列ベースのアクセス] を選択し、[列を含める] を選択します。
   c. `product_name`、`price`、`category` 列を選択します。
   d. [付与] を選択します。

Athena を使用した Delta Lake テーブルをクエリするには

ここで Athena を使用し、作成した Delta Lake テーブルに対するクエリを開始します。初めて Athena でクエリを実行する場合は、クエリ結果の場所を設定する必要があります。詳細については、「クエリ結果の場所の指定」を参照してください。

1. データレイク管理者ユーザーとしてログアウトし、 CloudFormation 出力から前に書き留めたパスワードを使用して、米国東部 (バージニア北部) リージョン`BusinessAnalystUser`で としてログインします。

2. https://console.aws.amazon.com/athena/ で Athena コンソールを開きます。
3. [設定] を選択し、[管理] を選択します。
4. クエリ結果の場所 ボックスに、 CloudFormation 出力で作成したバケットへのパスを入力します。`AthenaQueryResultLocation` (s3://lf-otf-tutorial-123456789012/athena-results/) の値をコピーして、[保存] します。
5. 次のクエリを実行して、Delta Lake テーブルに保存されている 10 個のレコードをプレビューします。

```
select * from lfdeltadb.product limit 10;
```

Delta Lake テーブルをクエリする方法の詳細については、「Amazon Athena ユーザーガイド」の「Delta Lake テーブルのクエリ」を参照してください。

ステップ 5: AWS リソースをクリーンアップする

リソースをクリーンアップするには

への不要な請求を防ぐには AWS アカウント、このチュートリアルで使用した AWS リソースを削除します。

1. IAM 管理者として https://console.aws.amazon.com/cloudformation の CloudFormation コンソールにサインインします。
2. CloudFormation スタックを削除します。作成したテーブルは、スタックと共に自動的に削除されます。

Lake Formation のタグベースのアクセスコントロールを使用したデータレイクの管理

何千ものお客様がペタバイト規模のデータレイクを構築しています AWS。これらのお客様の多くは AWS Lake Formation 、 を使用して、組織全体でデータレイクを簡単に構築および共有します。テーブルとユーザーの数が増えるに従って、データスチュワードや管理者は、大規模なデータレイクに対する許可を容易に管理する方法を模索しています。Lake Formation のタグベースのアクセスコントロール (LF-TBAC) は、データスチュワードが (データの分類とオントロジーに基づいて) LF タグを作成し、これをリソースにアタッチできるようにすることで、この問題を解決します。

LF-TBAC は、属性に基づいて許可を定義する認可戦略です。これらの属性は、Lake Formation で LF タグと呼ばれています。LF タグは、Data Catalog リソースと Lake Formation プリンシパルにアタッチできます。データレイク管理者は、LF タグを使用して、Lake Formation リソースに対する許可を割り当てたり取り消したりできます。詳細については、「Lake Formation のタグベースのアクセス制御」を参照してください。。

このチュートリアルでは、 AWS パブリックデータセットを使用して Lake Formation タグベースのアクセスコントロールポリシーを作成する方法を示します。さらに、Lake Formation のタグベースのアクセスポリシーが関連付けられているテーブル、データベース、列に対してクエリを実行する方法も示します。

LF-TBAC は、以下のユースケースに使用できます。

- データレイク管理者がアクセス権を付与する必要があるテーブルやプリンシパルが多数ある
- オントロジーに基づいてデータを分類し、分類に基づいて許可を付与したい
- データレイク管理者が、疎結合方式で許可を動的に割り当てることを希望している

LF-TBAC を使用して許可を設定するための高レベルのステップを以下に示します。

1. データスチュワードが、Confidential および Sensitive の 2 つの LF タグを使用してタグオントロジーを定義します。Confidential=True のデータは、アクセスコントロールが厳しくなります。Sensitive=True のデータは、アナリストによる特定の分析を必要とします。
2. データスチュワードは、複数の異なる許可レベルをデータエンジニアに割り当てることで、LF タグが異なる複数のテーブルを構築します。
3. データエンジニアは、tag_database および col_tag_database の 2 つのデータベースを構築します。tag_database 内のすべてのテーブルには Confidential=True が設定されます。col_tag_database 内のすべてのテーブルには Confidential=False が設定されます。col_tag_database 内のテーブルにある一部の列には、特定の分析ニーズに応じて Sensitive=True がタグ付けされます。
4. データエンジニアは、特定の式条件 (Confidential=True、Confidential=False、Sensitive=True) を持つテーブルに対する読み取り許可をアナリストに付与します。
5. この設定により、データアナリストは適切なデータを使用した分析の実行に集中できます。

トピック

- 対象者

- 前提条件
- ステップ 1: リソースをプロビジョニングする
- ステップ 2: データのロケーションを登録し、LF タグオントロジーを作成して、アクセス許可を付与する
- ステップ 3: Lake Formation のデータベースを作成する
- ステップ 4: テーブルの許可を付与する
- ステップ 5: Amazon Athena でクエリを実行して許可を検証する
- ステップ 6: AWS リソースをクリーンアップする

対象者

このチュートリアルは、データスチュワード、データエンジニア、データアナリストを対象としています。Lake Formation でアクセス許可を管理 AWS Glue Data Catalog および管理する場合、生成アカウント内のデータスチュワードは、それらがサポートする機能に基づいて機能所有権を持ち、さまざまなコンシューマー、外部組織、およびアカウントにアクセス権を付与できます。

次の表は、このチュートリアルで使用するロールのリストです。

ロール	説明
データスチュワード (管理者)	`lf-data-steward` ユーザーには以下のアクセス権があります。 • Data Catalog 内のすべてのリソースに対する読み取りアクセス権 • LF-タグを作成し、データエンジニアロールに関連付けて、他のプリンシパルに許可を付与できる
データエンジニア	`lf-data-engineer` ユーザーには以下のアクセス権があります。 • Data Catalog 内のすべてのリソースに対する完全な読み取り、書き込み、更新のアクセス権

ロール	説明
	• データレイクでのデータのロケーションの許可 • LF タグの関連付けと、Data Catalog への関連付けができる • LF タグをリソースにアタッチし、データスチュワードが作成したポリシーに基づいてプリンシパルにアクセス権を提供できる
データアナリスト	`lf-data-analyst` ユーザーには以下のアクセス権があります。 • Lake Formation のタグベースのアクセスポリシーで共有されるリソースへのきめ細かなアクセス権

前提条件

このチュートリアルを開始する前に、適切なアクセス許可を持つ管理ユーザーとしてサインインするために AWS アカウント 使用できる が必要です。詳細については、「初期設定 AWS タスクを完了する」を参照してください。

このチュートリアルでは、ユーザーが IAM に精通していることを前提としています。IAM については、「IAM ユーザーガイド」を参照してください。

ステップ 1: リソースをプロビジョニングする

このチュートリアルには、クイックセットアップ用の AWS CloudFormation テンプレートが含まれています。このテンプレートを参照し、ニーズに合わせてカスタマイズできます。テンプレートでは、この演習を実行するための 3 つの異なるロール (対象者 を参照) を作成し、nyc-taxi-data データセットをローカル Amazon S3 バケットにコピーします。

- Amazon S3 バケット
- 適切な Lake Formation 設定
- 適切な Amazon EC2 リソース
- 認証情報を持つ 3 つの IAM ロール

リソースを作成する

1. 米国東部 (バージニア北部) リージョンの https://console.aws.amazon.com/cloudformation で AWS CloudFormation コンソールにサインインします。
2. [Launch Stack] (スタックの起動) を選択します。
3. [Next] (次へ) を選択します。
4. [User Configuration] (ユーザーの設定) セクションで、3 つのロール (`DataStewardUserPassword`、`DataEngineerUserPassword`、`DataAnalystUserPassword`) のパスワードを入力します。
5. 最終ページの詳細を確認し、IAM リソースを作成する AWS CloudFormation 可能性があることを確認します。
6. [作成] を選択します。

   スタックの作成には、最大 5 分かかる場合があります。

> **Note**
>
> チュートリアルを完了したら、 でスタックを削除 CloudFormation して、引き続き料金が発生しないようにすることができます。スタックのイベントステータスで、リソースが正常に削除されていることを確認してください。

ステップ 2: データのロケーションを登録し、LF タグオントロジーを作成して、アクセス許可を付与する

このステップでは、データスチュワードユーザーが 2 つの LF タグ (`Confidential` と `Sensitive`) を使用してタグオントロジーを定義し、新しく作成した LF タグをリソースにアタッチすることを特定の IAM プリンシパルに許可します。

データのロケーションを登録し、LF タグオントロジーを定義する

1. データスチュワードユーザー (`lf-data-steward`) として最初のステップを実行し、Lake Formation で Amazon S3 と Data Catalog のデータを検証します。

   a. CloudFormation スタックのデプロイ時に使用したパスワード`lf-data-steward`を使用して、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールにサインインします。

b. ナビゲーションペインの [Permissions] (許可) で、[Administrative roles and tasks] (管理ロールおよびタスク) を選択します。

c. [データレイク管理者] セクションで、[追加] を選択します。

d. [管理者を追加] ページの [IAM ユーザーとロール] で、ユーザー `lf-data-steward` を選択します。

e. [Save] (保存) を選択し、`lf-data-steward` を Lake Formation 管理者として追加します。

2. 次に、IAM ベースのアクセスコントロールではなく、Lake Formation の許可を使用してカタログリソースを制御するように、Data Catalog 設定を更新します。

a. ナビゲーションペインの [管理] で、[データカタログの設定] を選択します。

b. [Use only IAM access control for new databases] (新しいデータベースには IAM アクセスコントロールのみを使用する) チェックボックスをオフにします。

c. [Use only IAM access control for new tables in new databases] (新しいデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) チェックボックスをオフにします。

d. [保存] をクリックします。

3. 次に、データレイクのデータのロケーションを登録します。

a. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。

b. [Register location] (ロケーションを登録) を選択します。

c. [ロケーションを登録] ページで、[Amazon S3 パス] に `s3://lf-tagbased-demo-`*`Account-ID`* と入力します。

d. [IAM role] (IAM ロール) は、デフォルト値 `AWSServiceRoleForLakeFormationDataAccess` のままにします。

e. アクセス許可モードとして [Lake Formation] を選択します。

f. [Register location] (ロケーションを登録) を選択します。

4. 次に、LF タグを定義してオントロジーを作成します。

a. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] を選択します。

b. [Add LF-Tag] (LF タグを追加) を選択します。

c. [Key] (キー) に「`Confidential`」と入力します。

d. [Values] (値) で、`True` と `False` を追加します。

e. [Add LF-tag] (LF タグを追加) を選択します。

f. 同じ手順を繰り返して、Sensitive という LF タグを作成し、値を True に設定します。

これで、この演習に必要なすべての LF タグが作成されました。

IAM ユーザーに許可を付与する

1. 次に、新しく作成した LF タグをリソースにアタッチすることを特定の IAM プリンシパルに許可します。

   a. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] を選択します。

   b. [LF タグのアクセス許可] セクションで、[アクセス許可の付与] を選択します。

   c. [アクセス許可のタイプ] で、[LF タグのキーと値のペアのアクセス許可] を選択します。

   d. [IAM users and roles] (IAM ユーザーおよびロール) を選択します。

   e. [IAM user and roles] (IAM ユーザーおよびロール) で、lf-data-engineer ロールを選択します。

   f. [LF タグのアクセス許可スコープ] で、値が True および False のキー Confidential と、値が True のキー key Sensitive を追加します。

   g. [アクセス許可] で、[アクセス許可] と [付与可能なアクセス許可] として [記述] と [関連付け] を選択します。

   h. [付与] を選択します。

2. 次に、Data Catalog および によって作成された基盤となる Amazon S3 バケットにデータベースを作成するアクセス許可lf-data-engineerを に付与します AWS CloudFormation。

   a. ナビゲーションペインの [管理] で、[管理ロールとタスク] を選択します。

   b. [Database creators] (データベース作成者) セクションで、[Grant] (付与) を選択します。

   c. [IAM users and roles] (IAM ユーザーおよびロール) で、lf-data-engineer ロールを選択します。

   d. [Catalog permissions] (カタログの許可) で、[Create database] (データベースを作成) を選択します。

   e. [付与] を選択します。

3. 次に、Amazon S3 バケット (s3://lf-tagbased-demo-*Account-ID*) に対する許可を lf-data-engineer ユーザーに付与します。

a. ナビゲーションペインの [Permissions] (許可) で [Data locations] (データのロケーション) を選択します。

b. [付与] を選択します。

c. [My account] (マイアカウント) を選択します。

d. [IAM users and roles] (IAM ユーザーおよびロール) で、`lf-data-engineer` ロールを選択します。

e. ストレージの場所には、 CloudFormation テンプレート によって作成された Amazon S3 バケットを入力します(`s3://lf-tagbased-demo-`*`Account-ID`*)。

f. [付与] を選択します。

4. 次に、`lf-data-engineer` に、LF タグ式 `Confidential=True` に関連付けられたリソースに対する付与可能なアクセス許可を付与します。

a. ナビゲーションペインの [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択します。

b. [付与] を選択します。

c. [IAM users and roles] (IAM ユーザーおよびロール) を選択します。

d. ロール `lf-data-engineer` を選択します。

e. [LF タグまたはカタログリソース] セクションで、[LF タグに一致するリソース] を選択します。

f. [LF タグのキーと値のペアを追加] を選択します。

g. 値が `True` のキー `Confidential` を追加します。

h. [Database permissions] (データベースの許可) セクションで、[Database permissions] (データベースの許可) と [Grantable permissions] (付与可能な許可) の [Describe] (記述) を選択します。

i. [テーブルのアクセス許可] セクションで、[テーブルの許可] と [付与可能なアクセス許可] の両方で [記述]、[選択]、[変更] を選択します。

j. [付与] を選択します。

5. 次に、`lf-data-engineer` に、LF タグ式 `Confidential=False` に関連付けられたリソースに対する付与可能なアクセス許可を付与します。

a. ナビゲーションペインの [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択します。

b. [付与] を選択します。

c. [IAM users and roles] (IAM ユーザーおよびロール) を選択します。

d. ロール `lf-data-engineer` を選択します。

e. [Resources matched by LF-tags] (LF タグに一致するリソース) を選択します。

f. [Add LF-tag] (LF タグを追加) を選択します。

g. 値が `False` のキー `Confidential` を追加します。

h. [Database permissions] (データベースの許可) セクションで、[Database permissions] (データベースの許可) と [Grantable permissions] (付与可能な許可) の [Describe] (記述) を選択します。

i. [Table and column permissions] (テーブルと列の許可) セクションでは、何も選択しません。

j. [付与] を選択します。

6. 次に、`lf-data-engineer` に、LF タグのキーと値のペア `Confidential=False` と `Sensitive=True` に関連付けられたリソースに対する付与可能なアクセス許可を付与します。

a. ナビゲーションペインの [Permissions] (許可) で [Data permissions] (データの許可) を選択します。

b. [付与] を選択します。

c. [IAM users and roles] (IAM ユーザーおよびロール) を選択します。

d. ロール `lf-data-engineer` を選択します。

e. [LF タグまたはカタログリソース] セクションで、[LF タグに一致するリソース] を選択します。

f. [Add LF-Tag] (LF タグを追加) を選択します。

g. 値が `False` のキー `Confidential` を追加します。

h. [LF タグのキーと値のペアを追加] を選択します。

i. 値が `True` のキー `Sensitive` を追加します。

j. [Database permissions] (データベースの許可) セクションで、[Database permissions] (データベースの許可) と [Grantable permissions] (付与可能な許可) の [Describe] (記述) を選択します。

k. [テーブルのアクセス許可] セクションで、[テーブルの許可] と [付与可能なアクセス許可] の両方で [記述]、[選択]、[変更] を選択します。

l. [付与] を選択します。

ステップ 3: Lake Formation のデータベースを作成する

このステップでは、2 つのデータベースを作成し、テスト目的でデータベースと特定の列に LF タグをアタッチします。

データベースレベルのアクセス用にデータベースとテーブルを作成する

1. 最初に、データベース `tag_database` とテーブル `source_data` を作成し、適切な LF タグをアタッチします。

   a. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) の [データカタログ] で、[データベース] を選択します。

   b. [データベースの作成] を選択します。

   c. [Name] (名前) に「`tag_database`」と入力します。

   d. Location には、 CloudFormation テンプレート によって作成された Amazon S3 の場所を入力します(`s3://lf-tagbased-demo-`*`Account-ID`*`/tag_database/`)。

   e. [Use only IAM access control for new tables in this database] (このデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) を選択解除します。

   f. [データベースの作成] を選択します。

2. 次に、`tag_database` 内に新しいテーブルを作成します。

   a. [Databases] (データベース) ページで、データベース `tag_database` を選択します。

   b. [View tables] (テーブルの表示) を選択し、[Create table] (テーブルを作成) をクリックします。

   c. [Name] (名前) に「`source_data`」と入力します。

   d. [Database] (データベース) で、データベース `tag_database` を選択します。

   e. Table format で、Standard AWS Glue table を選択します。

   f. [Data is located in] (データの場所) で、[Specified path in my account] (自分のアカウントで指定したパス) を選択します。

   g. 含めるパス に、 CloudFormation テンプレート によって`tag_database`作成された へのパスを入力します(`s3://lf-tagbased-demo`*`Account-ID`*`/tag_database/`)。

   h. [Data format] (データ形式) で、[CSV] を選択します。

   i. [Upload schema] (スキーマのアップロード) で、次の JSON 配列の列構造を入力してスキーマを作成します。

```
[
                {
                     "Name": "vendorid",
                     "Type": "string"
                },
                {
                     "Name": "lpep_pickup_datetime",
                     "Type": "string"
                },
                {
                     "Name": "lpep_dropoff_datetime",
                     "Type": "string"

                },
                   {
                     "Name": "store_and_fwd_flag",
                     "Type": "string"
                },
                   {
                     "Name": "ratecodeid",
                     "Type": "string"

                },
                   {
                     "Name": "pulocationid",
                     "Type": "string"

                },
                {
                     "Name": "dolocationid",
                     "Type": "string"

                },
                   {
                     "Name": "passenger_count",
                     "Type": "string"

                },
                {
                     "Name": "trip_distance",
                     "Type": "string"

                },
```

```
                {
                  "Name": "fare_amount",
                  "Type": "string"

                },
                {
                  "Name": "extra",
                  "Type": "string"

                },
                {
                  "Name": "mta_tax",
                  "Type": "string"

                },
                {
                  "Name": "tip_amount",
                  "Type": "string"

                },
                {
                  "Name": "tolls_amount",
                  "Type": "string"

                },
                {
                  "Name": "ehail_fee",
                  "Type": "string"

                },
                {
                  "Name": "improvement_surcharge",
                  "Type": "string"

                },
                {
                  "Name": "total_amount",
                  "Type": "string"

                },
                {
                  "Name": "payment_type",
                  "Type": "string"
```

```
                }
]
```

j. アップロード を選択します。スキーマをアップロードすると、テーブルスキーマは次のスクリーンショットのようになります。

#	Column Name	Data type
1	vendorid	string
2	lpep_pickup_datetime	string
3	lpep_dropoff_datetime	string
4	store_and_fwd_flag	string
5	ratecodeid	string
6	pulocationid	string
7	dolocationid	string
8	passenger_count	string
9	trip_distance	string
10	fare_amount	string
11	extra	string
12	mta_tax	string
13	tip_amount	string
14	tolls_amount	string
15	ehail_fee	string
16	improvement_surcharge	string
17	total_amount	string
18	payment_type	string

k. [Submit] を選択してください。

3. 次に、データベースレベルで LF タグをアタッチします。

a. [Databases] (データベース) ページで、tag_database を見つけて選択します。

b. [アクション] メニューで、[LF タグの編集] を選択します。

c. [Assign new LF-tag] (新しい LF タグを割り当てる) を選択します。

d. [割り当てられたキー] で、以前に作成した Confidential LF タグを選択します。

e. [Values] (値) で、True を選択します。

f. [Save] (保存) を選択します。

これで、tag_database データベースへの LF タグの割り当ては完了です。

列レベルのアクセス用にデータベースとテーブルを作成する

以下の手順を繰り返して、データベース col_tag_database とテーブル source_data_col_lvl を作成し、列レベルで LF タグをアタッチします。

1. [Databases] (データベース) ページで、[Create database] (データベースを作成) を選択します。
2. [Name] (名前) に「col_tag_database」と入力します。
3. Location には、 CloudFormation テンプレート によって作成された Amazon S3 の場所を入力します(s3://lf-tagbased-demo-*Account-ID*/col_tag_database/)。
4. [Use only IAM access control for new tables in this database] (このデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) を選択解除します。
5. [データベースの作成] を選択します。
6. [Databases] (データベース) ページで、新しいデータベース (col_tag_database) を選択します。
7. [テーブルを表示] を選択し、[テーブルを作成] をクリックします。
8. [名前] にsource_data_col_lvlと入力してください。
9. [Database] (データベース) で、新しいデータベース (col_tag_database) を選択します。
10. Table format で、Standard AWS Glue table を選択します。
11. [Data is located in] (データの場所) で、[Specified path in my account] (自分のアカウントで指定したパス) を選択します。

12. col_tag_database (s3://lf-tagbased-demo-*Account-ID*/col_tag_database/) に Amazon S3 パスを入力します。
13. [Data format] (データ形式) で、CSV を選択します。
14. Upload schema の下に、次のスキーマ JSON を入力します。

```
[
                {
                     "Name": "vendorid",
                     "Type": "string"


                },
                {
                     "Name": "lpep_pickup_datetime",
                     "Type": "string"


                },
                {
                     "Name": "lpep_dropoff_datetime",
                     "Type": "string"


                },
                   {
                     "Name": "store_and_fwd_flag",
                     "Type": "string"


                },
                   {
                     "Name": "ratecodeid",
                     "Type": "string"


                },
                   {
                     "Name": "pulocationid",
                     "Type": "string"


                },
```

```
                {
                     "Name": "dolocationid",
                     "Type": "string"


                },
                   {
                     "Name": "passenger_count",
                     "Type": "string"


                },
                {
                     "Name": "trip_distance",
                     "Type": "string"


                },
                   {
                     "Name": "fare_amount",
                     "Type": "string"


                },
                {
                     "Name": "extra",
                     "Type": "string"


                },
                   {
                     "Name": "mta_tax",
                     "Type": "string"


                },
                {
                     "Name": "tip_amount",
                     "Type": "string"


                },
                   {
                     "Name": "tolls_amount",
```

```
                    "Type": "string"


               },
               {
                    "Name": "ehail_fee",
                    "Type": "string"


               },
               {
                    "Name": "improvement_surcharge",
                    "Type": "string"


               },
               {
                    "Name": "total_amount",
                    "Type": "string"


               },
               {
                    "Name": "payment_type",
                    "Type": "string"


               }
]
```

15. Upload を選択してください。スキーマをアップロードすると、テーブルスキーマは次のスクリーンショットのようになります。

#	Column Name	Data type
1	vendorid	string
2	lpep_pickup_datetime	string
3	lpep_dropoff_datetime	string
4	store_and_fwd_flag	string
5	ratecodeid	string
6	pulocationid	string
7	dolocationid	string
8	passenger_count	string
9	trip_distance	string
10	fare_amount	string
11	extra	string
12	mta_tax	string
13	tip_amount	string
14	tolls_amount	string
15	ehail_fee	string
16	improvement_surcharge	string
17	total_amount	string
18	payment_type	string

16. [Submit] (送信) を選択して、テーブルの作成を完了します。
17. 次に、Sensitive=True LF タグを列 vendorid および fare_amount に関連付けます。

    a. [Tables] (テーブル) ページで、(source_data_col_lvl) で作成したテーブルを選択します。

    b. [アクション] メニューで、[スキーマ] を選択します。

    c. 列 vendorid を選択し、[LF タグの編集] を選択します。

    d. [Assigned keys] (割り当てられたキー) で、[Sensitive] (機密) を選択します。

    e. [Values] (値) で、True を選択します。

    f. [Save] (保存) を選択します。
18. 次に、Confidential=False LF タグを col_tag_database に関連付けます。これは、がログインcol_tag_database時にデータベースを記述できるようにするためlf-data-analystに必要です Amazon Athena。

    a. [Databases] (データベース) ページで、col_tag_database を見つけて選択します。

    b. [アクション] メニューで、[LF タグの編集] を選択します。

    c. [Assign new LF-Tag] (新しい LF タグを割り当てる) を選択します。

    d. [割り当てられたキー] で、以前に作成した Confidential LF タグを選択します。

    e. [Values] (値) で、False を選択します。

    f. [Save] (保存) を選択します。

ステップ 4: テーブルの許可を付与する

LF タグ Confidential および Sensitive を使用して、データベース tag_database とテーブル col_tag_database の使用許可をデータアナリストに付与します。

1. LF タグ Confidential=True (データベース: tag_database) に関連付けられたオブジェクトへのアクセス許可を lf-data-analyst ユーザーに付与して、データベースの Describe (記述) とテーブルの Select (選択) を許可するには、以下の手順に従います。

    a. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) に lf-data-engineer としてサインインします。

    b. [アクセス許可] で、[データレイクのアクセス許可] を選択します。

    c. [付与] を選択します。

d. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール) を選択します。

e. [IAM user and roles] (IAM ユーザーおよびロール) で、`lf-data-analyst` を選択します。

f. [LF タグまたはカタログリソース] で、[LF タグに一致するリソース] を選択します。

g. [Add LF-Tag] (LF タグを追加) を選択します。

h. [Key] (キー) で、`Confidential` を選択します。

i. [Values] (値) で、`True` を選択します。

j. [Database permissions] (データベースの許可) で、`Describe` を選択します。

k. [Table permissions] (テーブルの許可) で、[Select] (選択) と [Describe] (記述) を選択します。

l. [付与] を選択します。

2. 次に、同じ手順を繰り返して、LF タグ式 `Confidential=False` に対するアクセス許可をデータアナストに付与します。この LF タグは、Amazon Athena から `lf-data-analyst` として ログインしたときに、`col_tag_database` とテーブル `source_data_col_lvl` を記述するために使用します。

a. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) に `lf-data-engineer` としてサインインします。

b. [Databases] (データベース) ページで、データベース `col_tag_database` を選択します。

c. [Actions] (アクション)、[Grant] (付与) の順に選択します。

d. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール) を選択します。

e. [IAM user and roles] (IAM ユーザーおよびロール) で、`lf-data-analyst` を選択します。

f. [LF タグに一致するリソース] を選択します。

g. [Add LF-Tag] (LF タグを追加) を選択します。

h. [Key] (キー) で、`Confidential` を選択します。

i. [Values] (値) で、`False` を選択します。

j. [Database permissions] (データベースの許可) で、`Describe` を選択します。

k. [Table permissions] (テーブルの許可) では、何も選択しません。

l. [付与] を選択します。

3. 次に、同じ手順を繰り返して、LF タグ式 `Confidential=False` と `Sensitive=True` に対するアクセス許可をデータアナストに付与します。この LF タグは、Amazon Athena

から lf-data-analyst としてログインしたときに、col_tag_database とテーブル source_data_col_lvl (列レベル) を記述するために使用します。

a. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) に lf-data-engineer としてサインインします。
b. [Databases] (データベース) ページで、データベース col_tag_database を選択します。
c. [Actions] (アクション)、[Grant] (付与) の順に選択します。
d. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール) を選択します。
e. [IAM user and roles] (IAM ユーザーおよびロール) で、lf-data-analyst を選択します。
f. [LF タグに一致するリソース] を選択します。
g. [Add LF-Tag] (LF タグを追加) を選択します。
h. [Key] (キー) で、Confidential を選択します。
i. [Values] (値) で、False を選択します。
j. [Add LF-tag] (LF タグを追加) を選択します。
k. [Key] (キー) で、Sensitive を選択します。
l. [Values] (値) で、True を選択します。
m. [Database permissions] (データベースの許可) で、Describe を選択します。
n. [Table permissions] (テーブルの許可) で、Select と Describe を選択します。
o. [付与] を選択します。

ステップ 5: Amazon Athena でクエリを実行して許可を検証する

このステップでは、Amazon Athena を使用して 2 つのテーブル (source_data and source_data_col_lvl) に対して SELECT クエリを実行します。クエリ結果の場所 (s3://lf-tagbased-demo-*Account-ID*/athena-results/) として Amazon S3 パスを使用します。

1. Athena コンソール (https://console.aws.amazon.com/athena/) に lf-data-analyst としてサインインします。
2. Athena クエリエディタの左側のパネルで、tag_database を選択します。
3. source_data の横にある追加のメニューオプションアイコン (縦の 3 つのドット) を選択し、[Preview table] (テーブルのプレビュー) を選択します。
4. [Run query] (クエリの実行) を選択します。

クエリの実行には数分かかることがあります。このクエリでは、すべての列が出力に表示されます。LF タグがデータベースレベルで関連付けられていて、source_data テーブルはデータベース tag_database から LF-tag を自動的に継承しているためです。

5. col_tag_database と source_data_col_lvl を使用して別のクエリを実行します。

   2 番目のクエリは、Non-Confidential および Sensitive としてタグ付けされた 2 つの列を返します。

6. また、ポリシーの許可を持たない列に対する Lake Formation のタグベースのアクセスポリシーの動作を確認することもできます。テーブル source_data_col_lvl からタグなしの列を選択すると、Athena はエラーを返します。例えば、次のクエリを実行すると、タグなしの列 geolocationid が選択されます。

```
SELECT geolocationid FROM "col_tag_database"."source_data_col_lvl" limit 10;
```

ステップ 6: AWS リソースをクリーンアップする

に不要な料金が発生しないように AWS アカウント、このチュートリアルで使用した AWS リソースを削除できます。

1. Lake Formation コンソールに lf-data-engineer としてサインインし、データベース tag_database および col_tag_database を削除します。
2. 次に、lf-data-steward としてサインインし、上で lf-data-engineer および lf-data-analyst. に対して付与したすべての LF タグの許可、データの許可、データのロケーションの許可を消去します。
3. CloudFormation スタックのデプロイに使用した IAM 認証情報を使用して、アカウント所有者として Amazon S3 コンソールにサインインします。
4. 以下のバケットを削除します。

   - lf-tagbased-demo-accesslogs-*acct-id*
   - lf-tagbased-demo-*acct-id*

5. https://console.aws.amazon.com/cloudformation で CloudFormation コンソールにサインインし、作成したスタックを削除します。スタックステータスが DELETE_COMPLETE に変わるまで待ちます。

行レベルのアクセスコントロールによるデータレイクの保護

AWS Lake Formation 行レベルのアクセス許可を使用すると、データコンプライアンスとガバナンスポリシーに基づいて、テーブル内の特定の行へのアクセスを提供できます。数十億のレコードを格納する大きなテーブルがある場合、さまざまなユーザーやチームがアクセスして表示できるデータを、許可した範囲に限定する方法が必要です。行レベルのアクセスコントロールは、データを保護するとともに、ジョブの実行に必要なデータへのアクセス許可をユーザーに付与するシンプルでパフォーマンスの高い方法です。Lake Formation は、一元的な監査とコンプライアンスレポートを通じて、どのプリンシパルが、どのデータに、いつ、どのサービスを通じてアクセスしたかを特定します。

このチュートリアルでは、Lake Formation での行レベルのアクセスコントロールの仕組みと設定方法について説明します。

このチュートリアルには、必要なリソースをすばやくセットアップするための AWS CloudFormation テンプレートが含まれています。このテンプレートを参照し、ニーズに合わせてカスタマイズできます。

トピック

- 対象者
- 前提条件
- ステップ 1: リソースをプロビジョニングする
- ステップ 2: データフィルターなしでクエリを実行する
- ステップ 3: データフィルターを設定し、許可を付与する
- ステップ 4: データフィルターを使用してクエリを実行する
- ステップ 5: AWS リソースをクリーンアップする

対象者

このチュートリアルは、データスチュワード、データエンジニア、データアナリストを対象としています。次の表は、データ所有者とデータコンシューマーのロールと責任を示しています。

ロール	説明
IAM 管理者	ユーザーおよびロール、Amazon Simple Storage Service (Amazon S3) バケットを作成

ロール	説明
	できるユーザー。AdministratorAccess AWS 管理ポリシーがあります。
データレイク管理者	データレイクの設定、データフィルターの作成、およびデータアナリストへの許可の付与を担当するユーザー。
データアナリスト	データレイクに対してクエリを実行できるユーザー。複数の異なる国 (このユースケースの場合は日本と米国) に居住するデータアナリストは、自国の顧客の製品レビューのみを分析でき、コンプライアンス上の理由から、他国の顧客のデータを表示することはできません。

前提条件

このチュートリアルを開始する前に、適切なアクセス許可を持つ管理ユーザーとしてサインインするために AWS アカウント 使用できる が必要です。詳細については、「初期設定 AWS タスクを完了する」を参照してください。

このチュートリアルでは、ユーザーが IAM に精通していることを前提としています。IAM については、「IAM ユーザーガイド」を参照してください。

Lake Formation 設定を変更する

> **Important**
>
> CloudFormation テンプレートを起動する前に、以下の手順に従って、Lake Formation の新しいデータベース/テーブルの IAM アクセスコントロールのみを使用する オプションを無効にします。

1. 米国東部 (バージニア北部) リージョンまたは米国西部 (オレゴン) リージョンで、Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインします。
2. [Data Catalog] で、[Settings] (設定) を選択します。

3. [Use only IAM access control for new databases] (新しいデータベースには IAM アクセスコントロールのみを使用する) と [Use only IAM access control for new tables in new databases] (新しいデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) を選択解除します。
4. [Save] (保存) を選択します。

ステップ 1: リソースをプロビジョニングする

このチュートリアルには、クイックセットアップ用の CloudFormation テンプレートが含まれています。このテンプレートを参照し、ニーズに合わせてカスタマイズできます。 CloudFormation テンプレートは、次のリソースを生成します。

- ユーザーおよびポリシー (以下のロール向け):
  - DataLakeAdmin
  - DataAnalystUS
  - DataAnalystJP
- Lake Formation データレイクの設定と許可
- サンプルデータファイルをパブリック Amazon S3 バケットから Amazon S3 バケットにコピーするために使用される Lambda 関数 (Lambda-backed CloudFormation カスタムリソース用)
- データレイクとして機能する Amazon S3 バケット
- AWS Glue Data Catalog データベース、テーブル、パーティション

リソースを作成する

CloudFormation テンプレートを使用してリソースを作成するには、次の手順に従います。

1. 米国東部 (バージニア北部) リージョンの https://console.aws.amazon.com/cloudformation で CloudFormation コンソールにサインインします。
2. [スタックの起動] を選択します。
3. [Create Stack] (スタックの作成) 画面で、[Next] (次へ) を選択します。
4. [Stack name] (スタック名) を入力します。
5. [DataLakeAdminUserName] と [DataLakeAdminUserPassword] に、データレイク管理者ユーザーとして自分の IAM ユーザーネームおよびパスワードを入力します。

6. [DataAnalystUsUserName] と [DataAnalystUsUserPassword] に、米国マーケットプレイスを担当するデータアナリストとして指定するユーザーのユーザーネームとパスワードを入力します。
7. [DataAnalystJpUserName] と [DataAnalystJpUserPassword] に、日本マーケットプレイスを担当するデータアナリストとして指定するユーザーのユーザーネームとパスワードを入力します。
8. [DataLakeBucketName] に、データバケットの名前を入力します。
9. [DatabaseName] と [TableName] は、デフォルトのままにします。
10. [Next] (次へ) を選択します。
11. 次のページで、[Next] (次へ) を選択します。
12. 最終ページの詳細を確認し、IAM リソースを作成する CloudFormation 可能性があることを確認します。
13. [作成] を選択します。

    スタックの作成が完了するまでに 1 分かかる場合があります。

ステップ 2: データフィルターなしでクエリを実行する

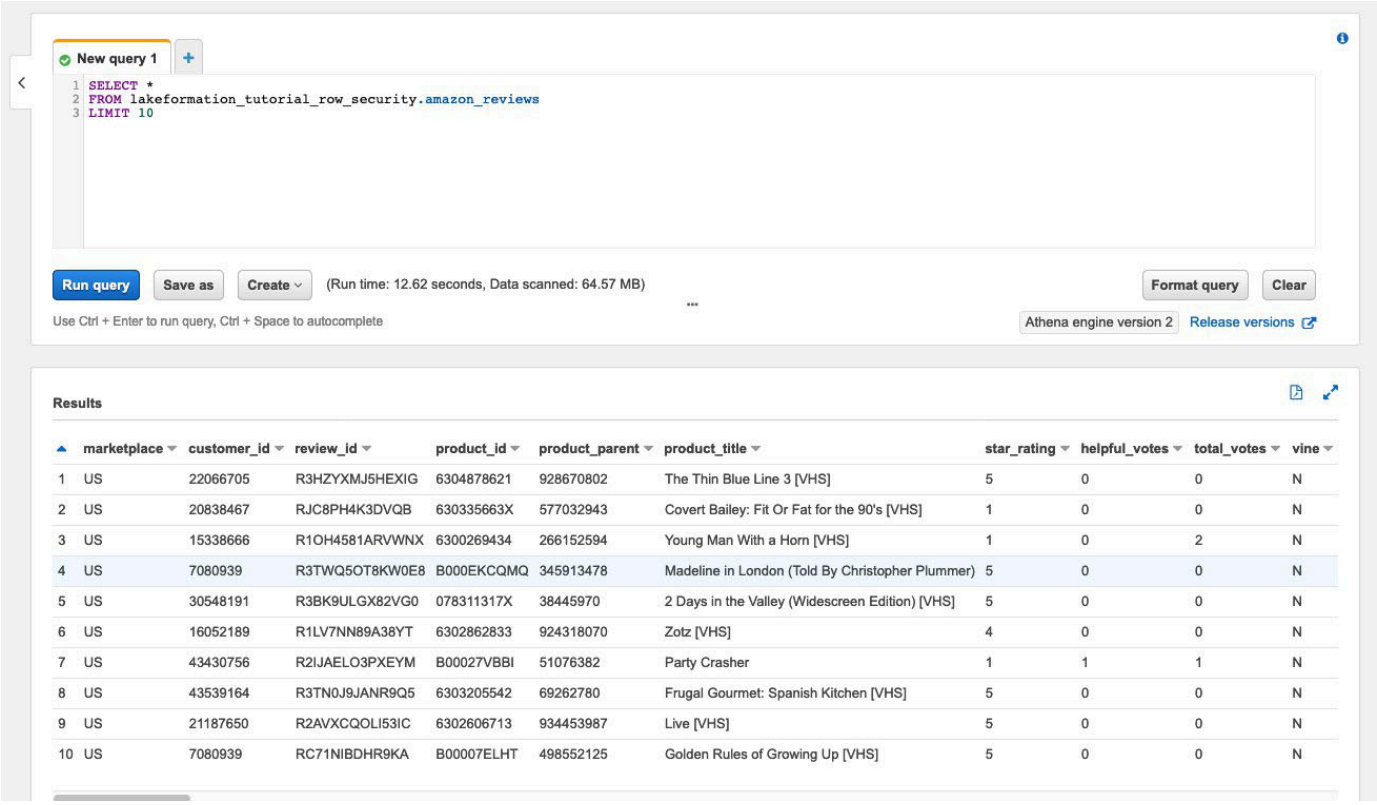
環境の設定後に、製品レビューテーブルに対してクエリを実行できます。まず、行レベルのアクセスコントロールなしでテーブルにクエリを実行し、データが表示されることを確認します。Amazon Athena でクエリを初めて実行する場合は、クエリ結果の場所を設定する必要があります。

行レベルのアクセスコントロールなしでテーブルに対してクエリを実行する

1. `DatalakeAdmin` ユーザーとして Athena コンソール (https://console.aws.amazon.com/athena/) にサインインし、次のクエリを実行します。

   ```
   SELECT *
   FROM lakeformation_tutorial_row_security.amazon_reviews
   LIMIT 10
   ```

   次のスクリーンショットは、クエリ結果を示しています。このテーブルのパーティションは 1 つ (`product_category=Video`) のみであるため、各レコードは動画製品のレビューコメントを示します。

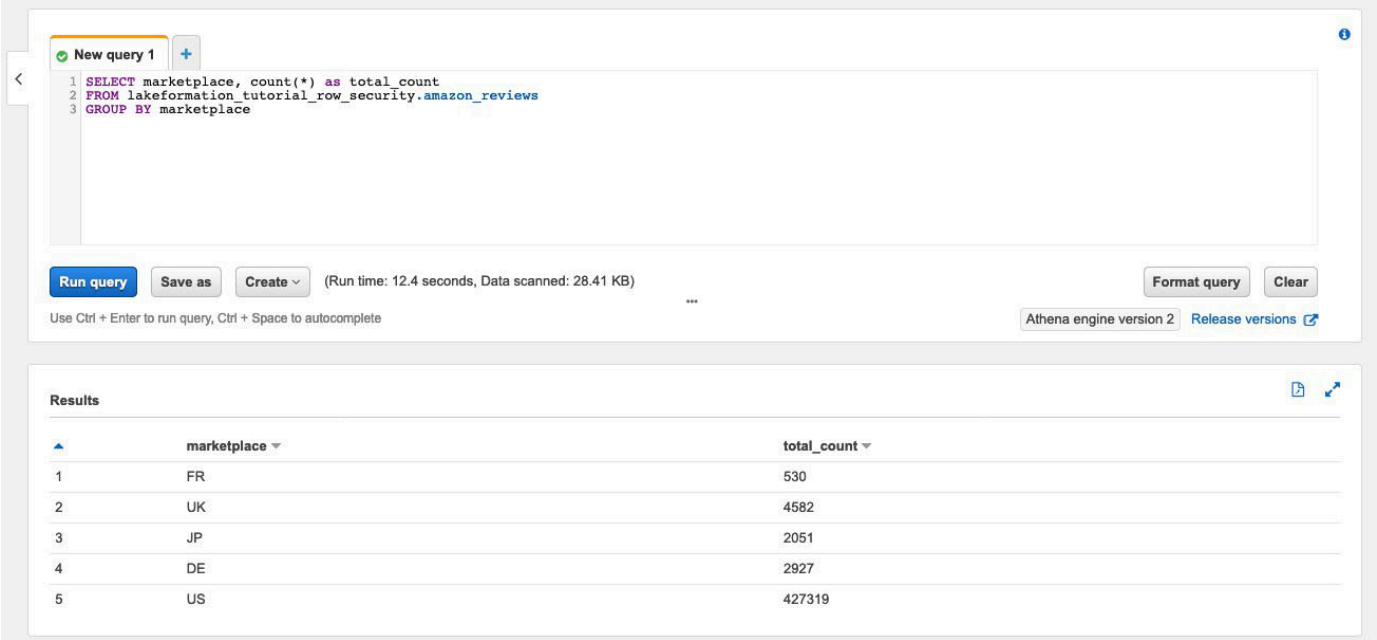


	marketplace	customer_id	review_id	product_id	product_parent	product_title	star_rating	helpful_votes	total_votes	vine
1	US	22066705	R3HZYXMJ5HEXIG	6304878621	928670802	The Thin Blue Line 3 [VHS]	5	0	0	N
2	US	20838467	RJC8PH4K3DVQB	630335663X	577032943	Covert Bailey: Fit Or Fat for the 90's [VHS]	1	0	0	N
3	US	15338666	R1OH4581ARVWNX	6300269434	266152594	Young Man With a Horn [VHS]	1	0	2	N
4	US	7080939	R3TWQ5OT8KW0E8	B000EKCQMQ	345913478	Madeline in London (Told By Christopher Plummer)	5	0	0	N
5	US	30548191	R3BK9ULGX82VG0	078311317X	38445970	2 Days in the Valley (Widescreen Edition) [VHS]	5	0	0	N
6	US	16052189	R1LV7NN89A38YT	6302862833	924318070	Zotz [VHS]	4	0	0	N
7	US	43430756	R2IJAELO3PXEYM	B00027VBBI	51076382	Party Crasher	1	1	1	N
8	US	43539164	R3TN0J9JANR9Q5	6303205542	69262780	Frugal Gourmet: Spanish Kitchen [VHS]	5	0	0	N
9	US	21187650	R2AVXCQOLI53IC	6302606713	934453987	Live [VHS]	5	0	0	N
10	US	7080939	RC71NIBDHR9KA	B00007ELHT	498552125	Golden Rules of Growing Up [VHS]	5	0	0	N

2. 次に、集計クエリを実行して、marketplace あたりのレコードの総数を取得します。

```
SELECT marketplace, count(*) as total_count
FROM lakeformation_tutorial_row_security.amazon_reviews
GROUP BY marketplace
```

次のスクリーンショットは、クエリ結果を示しています。marketplace 列には 5 つの異なる値があります。以降のステップでは、marketplace 列を使用して行ベースのフィルターをセットアップします。



ステップ 3: データフィルターを設定し、許可を付与する

このチュートリアルでは、2 人のデータアナリストを使用します。1 人は米国マーケットプレイス、もう 1 人は日本マーケットプレイスを担当しています。各アナリストは Athena を使用して、各担当マーケットプレイスのみのカスタマーレビューを分析します。2 つの異なるデータフィルターを作成します。1 つは米国マーケットプレイスを担当するアナリスト用、もう 1 つは日本マーケットプレイスを担当するアナリスト用です。次に、アナリストにそれぞれの許可を付与します。

データフィルターを作成して許可を付与する

1. US `marketplace` データへのアクセスを制限するためのフィルターを作成します。

   a. 米国東部 (バージニア北部) リージョンで `DatalakeAdmin` ユーザーとして Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインします。

   b. [Data filters] (データフィルター) を選択します。

   c. [Create new filter] (新しいフィルターの作成) を選択します。

   d. [Data filter name] (データフィルター名) に、「`amazon_reviews_US`」と入力します。

   e. [Target database] (ターゲットデータベース) で、データベース `lakeformation_tutorial_row_security` を選択します。

   f. [Target table] (ターゲットテーブル) で、テーブル `amazon_reviews` を選択します。

   g. [Column-level access] (列レベルのアクセス) は、デフォルトのままにします。

h. [Row filter expression] (行フィルター式) に「`marketplace='US'`」と入力します。

i. [フィルターを作成] をクリックします。

2. 日本の `marketplace` データへのアクセスを制限するフィルターを作成します。

a. [Data filters] (データフィルター) ページで、[Create new filter] (新しいフィルターを作成) を選択します。

b. [Data filter name] (データフィルター名) に、「`amazon_reviews_JP`」と入力します。

c. [Target database] (ターゲットデータベース) で、データベース `lakeformation_tutorial_row_security` を選択します。

d. [Target table] (ターゲットテーブル) で、`table amazon_reviews` を選択します。

e. [Column-level access] (列レベルのアクセス) は、デフォルトのままにします。

f. [Row filter expression] (行フィルター式) に「`marketplace='JP'`」と入力します。

g. [フィルターを作成] をクリックします。

3. 次に、これらのデータフィルターを使用して、データアナリストに許可を付与します。米国のデータアナリスト (`DataAnalystUS`) に許可を付与するには、以下のステップに従います。

a. [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択します。

b. [Data permission] (データの許可) で、[Grant] (付与) を選択します。

c. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール)を選択し、ロール `DataAnalystUS` を選択します。

d. [LF-tags or catalog resources] (LF タグまたはカタログリソース) で、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。

e. [Database] (データベース) で、`lakeformation_tutorial_row_security` を選択します。

f. [Tables-optional] (テーブル-オプション) で、`amazon_reviews` を選択します。

g. [Data filters – optional] (データフィルター - オプション) で、`amazon_reviews_US` を選択します。

h. [Data filter permissions] (データフィルターの許可) で、[Select] (選択) を選択します。

i. [付与] を選択します。

4. 日本のデータアナリスト (`DataAnalystJP`) に許可を付与するには、以下のステップに従います。

a. [Permissions] (許可) で、[Data lake permissions] (データレイクの許可) を選択します。

b. [Data permission] (データの許可) で、[Grant] (付与) を選択します。

c. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール)を選択し、ロール `DataAnalystJP` を選択します。

d. [LF-tags or catalog resources] (LF タグまたはカタログリソース) で、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。

e. [Database] (データベース) で、`lakeformation_tutorial_row_security` を選択します。

f. [Tables-optional] (テーブル-オプション) で、`amazon_reviews` を選択します。

g. [Data filters – optional] (データフィルター - オプション) で、`amazon_reviews_JP` を選択します。

h. [Data filter permissions] (データフィルターの許可) で、[Select] (選択) を選択します。

i. [付与] を選択します。

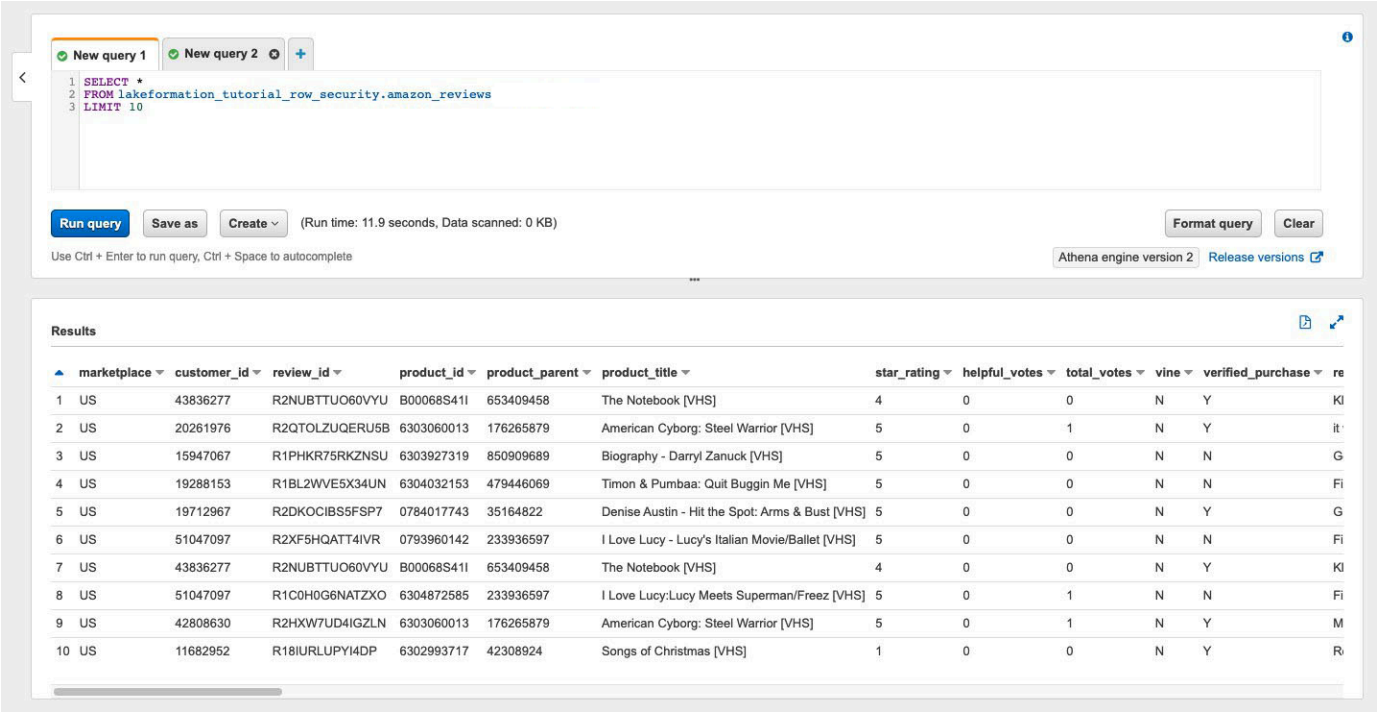
ステップ 4: データフィルターを使用してクエリを実行する

製品レビューテーブルにデータフィルターをアタッチして、いくつかのクエリを実行し、Lake Formation で許可がどのように適用されるかを確認します。

1. Athena コンソール (https://console.aws.amazon.com/athena/) に `DataAnalystUS` ユーザーとしてサインインします。
2. 次のクエリを実行し、定義した行レベルの許可に基づいてフィルタリングされたレコードをいくつか取得します。

```
SELECT *
FROM lakeformation_tutorial_row_security.amazon_reviews
LIMIT 10
```

次のスクリーンショットは、クエリ結果を示しています。



	marketplace	customer_id	review_id	product_id	product_parent	product_title	star_rating	helpful_votes	total_votes	vine	verified_purchase	re
1	US	43836277	R2NUBTTUO60VYU	B00068S41I	653409458	The Notebook [VHS]	4	0	0	N	Y	KI
2	US	20261976	R2QTOLZUQERU5B	6303060013	176265879	American Cyborg: Steel Warrior [VHS]	5	0	1	N	Y	it
3	US	15947067	R1PHKR75RKZNSU	6303927319	850909689	Biography - Darryl Zanuck [VHS]	5	0	0	N	N	G
4	US	19288153	R1BL2WVE5X34UN	6304032153	479446069	Timon & Pumbaa: Quit Buggin Me [VHS]	5	0	0	N	N	Fi
5	US	19712967	R2DKOCIBS5FSP7	0784017743	35164822	Denise Austin - Hit the Spot: Arms & Bust [VHS]	5	0	0	N	Y	G
6	US	51047097	R2XF5HQATT4IVR	0793960142	233936597	I Love Lucy - Lucy's Italian Movie/Ballet [VHS]	5	0	0	N	N	Fi
7	US	43836277	R2NUBTTUO60VYU	B00068S41I	653409458	The Notebook [VHS]	4	0	0	N	Y	KI
8	US	51047097	R1C0H0G6NATZXO	6304872585	233936597	I Love Lucy:Lucy Meets Superman/Freez [VHS]	5	0	1	N	N	Fi
9	US	42808630	R2HXW7UD4IGZLN	6303060013	176265879	American Cyborg: Steel Warrior [VHS]	5	0	1	N	Y	M
10	US	11682952	R18IURLUPYI4DP	6302993717	42308924	Songs of Christmas [VHS]	1	0	0	N	Y	R

3. 同様に、クエリを実行し、マーケットプレイスごとのレコードの総数をカウントします。

```
SELECT marketplace , count ( * ) as total_count
FROM lakeformation_tutorial_row_security .amazon_reviews
GROUP BY marketplace
```

このクエリ結果には、結果内の marketplace US のみが表示されます。これは、ユーザーに許可された表示は、marketplace 列の値が US と等しい行のみであるためです。

4. DataAnalystJP ユーザーに切り替えて、同じクエリを実行します。

```
SELECT *
FROM lakeformation_tutorial_row_security.amazon_reviews
LIMIT 10
```

クエリ結果には、JP marketplace に属するレコードのみが表示されます。

5. クエリを実行し、marketplace あたりのレコードの総数をカウントします。

```
SELECT marketplace, count(*) as total_count
FROM lakeformation_tutorial_row_security.amazon_reviews
GROUP BY marketplace
```

クエリ結果には、JP marketplace に属する行のみが表示されます。

ステップ 5: AWS リソースをクリーンアップする

リソースをクリーンアップする

に不要な料金が発生しないように AWS アカウント、このチュートリアルで使用した AWS リソースを削除できます。

- Cloud Formation スタックを削除します。

Lake Formation のタグベースのアクセスコントロールと名前付きリソースを使用したデータレイクの共有

このチュートリアルでは、データベース全体をコピーすることなく、データレイク内に保存されたデータを複数の企業、組織、またはビジネスユニットと安全に共有 AWS Lake Formation するように を設定する方法を示します。Lake Formation クロスアカウントアクセスコントロールを使用してデータベースとテーブルを別の と共有するには AWS アカウント 、次の 2 つのオプションがあります。

- Lake Formation のタグベースのアクセスコントロール (推奨)

  Lake Formation のタグベースのアクセスコントロールは、属性に基づいて許可を定義する認可戦略です。これらの属性は、Lake Formation で LF タグと呼ばれています。詳細については、「Lake Formation のタグベースのアクセスコントロールを使用したデータレイクの管理」を参照してください。

- Lake Formation の名前付きリソース

  Lake Formation の名前付きリソース方式は、リソースの許可を定義する認可戦略です。リソースには、データベース、テーブル、列が含まれます。データレイク管理者は、Lake Formation のリソースに対する許可を割り当てたり、取り消したりできます。詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

  データレイク管理者がリソースごとに許可を明示的に付与することを希望する場合は、名前付きリソースを使用することをお勧めします。名前付きリソースメソッドを使用して Data Catalog リソースに対する Lake Formation アクセス許可を外部アカウントに付与すると、Lake Formation は AWS Resource Access Manager (AWS RAM) を使用してリソースを共有します。

トピック

- 対象者
- プロデューサーアカウントで Lake Formation の Data Catalog 設定を構成する
- ステップ 1: AWS CloudFormation テンプレートを使用してリソースをプロビジョニングする
- ステップ 2: Lake Formation クロスアカウント共有の前提条件
- ステップ 3: タグベースのアクセスコントロール方式を使用してクロスアカウント共有を実装する
- ステップ 4: 名前付きリソース方式を実装する
- ステップ 5: AWS リソースをクリーンアップする

対象者

このチュートリアルは、データスチュワード、データエンジニア、データアナリストを対象としています。Lake Formation から Data Catalog テーブルを共有 AWS Glue し、Lake Formation でアクセス許可を管理する場合、生成アカウント内のデータスチュワードは、サポートする関数に基づいて機能所有権を持ち、さまざまなコンシューマー、外部組織、およびアカウントへのアクセスを許可できます。次の表は、このチュートリアルで使用するロールのリストです。

ロール	説明
DataLakeAdminProducer	データレイク管理者 IAM ユーザーには、以下のアクセス権があります。 • Data Catalog 内のすべてのリソースに対する完全な読み取り、書き込み、更新のアクセス権 • リソースへの許可を付与できる • 共有テーブルへのリソースリンクを作成できる • LF タグをリソースにアタッチし、データスチュワードが作成したポリシーに基づいてプリンシパルにアクセス権を付与できる
DataLakeAdminConsumer	データレイク管理者 IAM ユーザーには、以下のアクセス権があります。

ロール	説明
	• Data Catalog 内のすべてのリソースに対する完全な読み取り、書き込み、更新のアクセス権 • リソースへの許可を付与できる • 共有テーブルへのリソースリンクを作成できる • LF タグをリソースにアタッチし、データスチュワードが作成したポリシーに基づいてプリンシパルにアクセス権を付与できる
DataAnalyst	DataAnalyst ユーザーには、以下のアクセス権があります。 • Lake Formation のタグベースのアクセスポリシーまたは名前付きリソース方式を使用して共有しているリソースへのきめ細かなアクセス権

プロデューサーアカウントで Lake Formation の Data Catalog 設定を構成する

このチュートリアルを開始する前に、適切なアクセス許可を持つ管理ユーザーとしてサインインするために AWS アカウント 使用できる が必要です。詳細については、「初期設定 AWS タスクを完了する」を参照してください。

このチュートリアルでは、ユーザーが IAM に精通していることを前提としています。IAM については、「IAM ユーザーガイド」を参照してください。

プロデューサーアカウントで Lake Formation の Data Catalog 設定を構成する

> **Note**
>
> このチュートリアルでは、ソーステーブルを持つアカウントをプロデューサーアカウントと呼び、ソーステーブルにアクセスする必要があるアカウントをコンシューマーアカウントと呼びます。

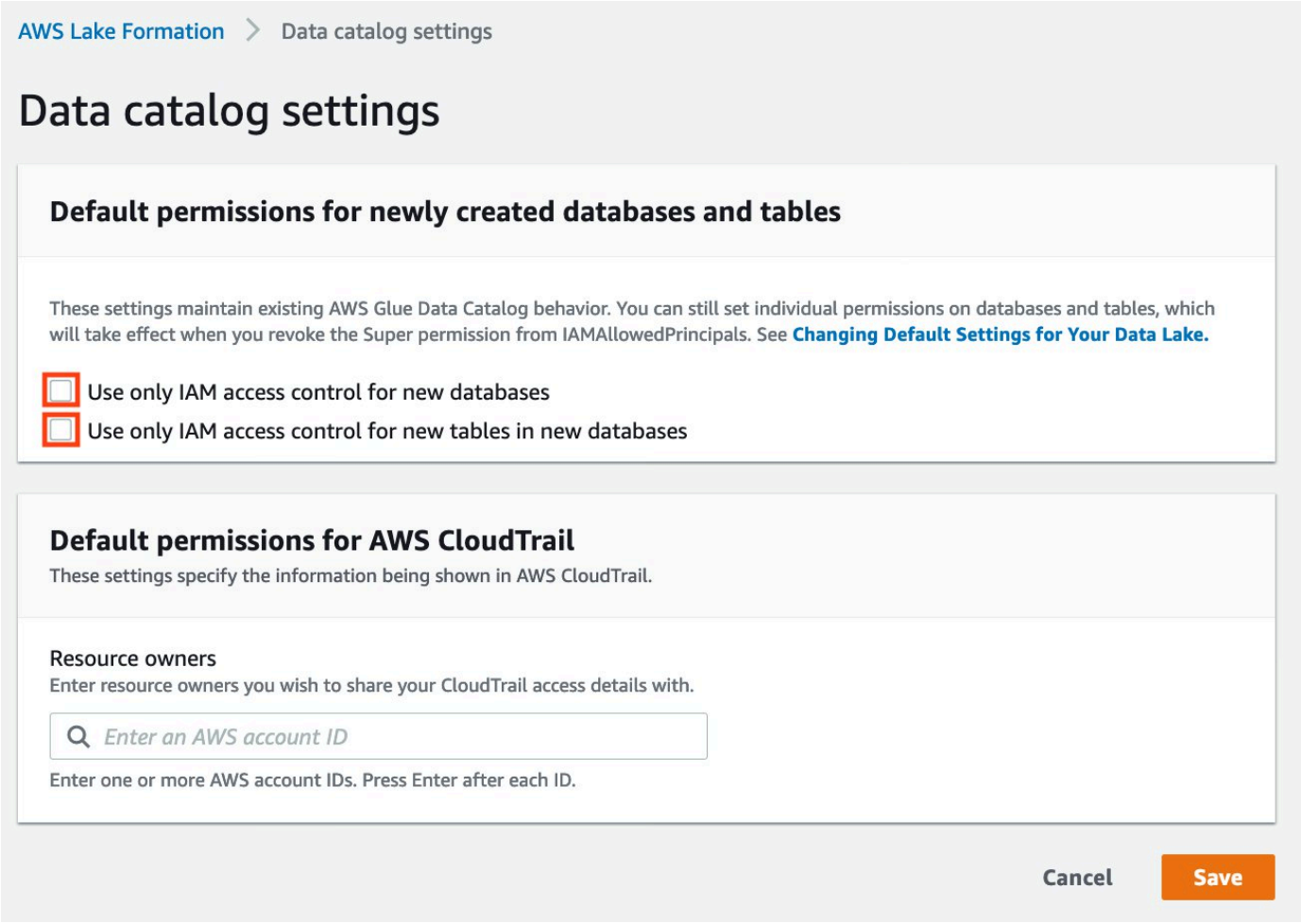
Lake Formation には、独自の許可管理モデルがあります。IAM アクセス許可モデルとの下位互換性を維持するために、アクセスSuper許可はデフォルトで既存のすべての AWS Glue Data Catalog リソースIAMAllowedPrincipalsの グループに付与されます。また、新しい Data Catalog リソースに対しては、[Use only IAM access control settings] (IAM アクセスコントロール設定のみを使用する) が有効になります。このチュートリアルでは、きめ細かなアクセスコントロールには Lake Formation の許可を使用し、きめの粗いアクセスコントロールには IAM ポリシーを使用します。詳細については、「細粒度のアクセスコントロールのための方式」を参照してください。したがって、クイックセットアップに AWS CloudFormation テンプレートを使用する前に、プロデューサーアカウントの Lake Formation Data Catalog 設定を変更する必要があります。

> **Important**
>
> この設定は、新しく作成したすべてのデータベースとテーブルに影響するため、このチュートリアルは非運用アカウントまたは新しいアカウントで実行することを強くお勧めします。また、共有アカウント (自社の開発アカウントなど) を使用している場合は、他のリソースに影響を与えないことを確認してください。デフォルトのセキュリティ設定を維持したい場合は、他のアカウントとリソースを共有するときに追加のステップを実行し、データベースやテーブルに対するデフォルトの Super 許可を IAMAllowedPrincipals から取り消す必要があります。詳細については、このチュートリアルの後半で説明します。

プロデューサーアカウントで Lake Formation の Data Catalog 設定を構成するには、以下のステップを実行します。

1. プロデューサーアカウント AWS マネジメントコンソール を使用して、管理者ユーザーまたは Lake Formation PutDataLakeSettings API アクセス許可を持つユーザーとして にサインインします。
2. Lake Formation コンソールのナビゲーションペインで、[Data Catalog] の [Settings] (設定) を選択します。
3. [Use only IAM access control for new databases] (新しいデータベースには IAM アクセスコントロールのみを使用する) と [Use only IAM access control for new tables in new databases] (新しいデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) を選択解除します。

   [保存] を選択します。



さらに、[Administrative roles and tasks] (管理ロールおよびタスク) の [Database creators] (データベース作成者) で、`IAMAllowedPrincipals` への `CREATE_DATABASE` 許可を削除できます。この後にのみ、Lake Formation の許可を使用して誰が新しいデータベースを作成できるかを管理できます。

ステップ 1: AWS CloudFormation テンプレートを使用してリソースをプロビジョニングする

プロデューサーアカウント用の CloudFormation テンプレートでは、以下のリソースを生成します。

- データレイクとなる Amazon S3 バケット。
- Lambda 関数 (Lambda-backed CloudFormation カスタムリソース用) 。この関数を使用して、パブリック Amazon S3 バケットからユーザーの Amazon S3 バケットにサンプルデータファイルをコピーします。

- IAM ユーザーおよびポリシー: DataLakeAdminProducer。
- Lake Formation の適切な設定および許可 (以下を含む):
  - プロデューサーアカウントで Lake Formation データレイク管理者を定義する
  - Amazon S3 バケットを Lake Formation データレイクのロケーションとして登録する (プロデューサーアカウント)
- AWS Glue Data Catalog データベース、テーブル、パーティション。間でリソースを共有するには 2 つのオプションがあるため AWS アカウント、このテンプレートはデータベースとテーブルの 2 つの別々のセットを作成します。

コンシューマーアカウントの CloudFormation テンプレートは、次のリソースを生成します。

- IAM ユーザーおよびポリシー:
  - DataLakeAdminConsumer
  - DataAnalyst
- AWS Glue Data Catalog データベース。このデータベースを使用して、共有リソースへのリソースリンクを作成します。

プロデューサーアカウントでリソースを作成する

1. 米国東部 (バージニア北部) リージョンの https://console.aws.amazon.com/cloudformation で AWS CloudFormation コンソールにサインインします。
2. [Launch Stack] (スタックの起動) を選択します。
3. [Next] (次へ) を選択します。
4. [Stack name] (スタック名) にスタック名 (`stack-producer` など) を入力します。
5. [User Configuration] (ユーザー設定) セクションで、[`ProducerDatalakeAdminUserName`] と [`ProducerDatalakeAdminUserPassword`] にユーザーネームとパスワードを入力します。
6. [DataLakeBucketName] に、データレイクバケットの名前を入力します。この名前はグローバルに一意である必要があります。
7. [DatabaseName] と [TableName] は、デフォルト値のままにします。
8. [Next] (次へ) を選択します。
9. 次のページで、[Next] (次へ) を選択します。
10. 最終ページの詳細を確認し、IAM リソースを作成する AWS CloudFormation 可能性があることを確認します。

11. [作成] を選択します。

    スタックの作成には、最大 1 分かかる場合があります。

コンシューマーアカウントでリソースを作成する

1. 米国東部 (バージニア北部) リージョンの https://console.aws.amazon.com/cloudformation で AWS CloudFormation コンソールにサインインします。
2. [Launch Stack] (スタックの起動) を選択します。
3. [Next] (次へ) を選択します。
4. [Stack name] (スタック名) にスタック名 (`stack-consumer` など) を入力します。
5. [User Configuration] (ユーザー設定) セクションで、[`ConsumerDatalakeAdminUserName`] と [`ConsumerDatalakeAdminUserPassword`] にユーザーネームとパスワードを入力します。
6. [`DataAnalystUserName`] と [`DataAnalystUserPassword`] に、データアナリスト IAM ユーザーを指定するユーザーネームとパスワードを入力します。
7. [DataLakeBucketName] に、データレイクバケットの名前を入力します。この名前はグローバルに一意である必要があります。
8. [DatabaseName] は、デフォルト値のままにします。
9. [`AthenaQueryResultS3BucketName`] に、Amazon Athena のクエリ結果を保存する Amazon S3 バケットの名前を入力します。バケットがない場合は、Amazon S3 バケットを作成します。
10. [Next] (次へ) を選択します。
11. 次のページで、[Next] (次へ) を選択します。
12. 最終ページの詳細を確認し、IAM リソースを作成する AWS CloudFormation 可能性があることを確認します。
13. [作成] を選択します。

    スタックの作成には、最大 1 分かかる場合があります。

> **Note**
>
> チュートリアルを完了したら、 でスタックを削除 CloudFormation して、料金が発生しないようにします。リソースが正常に削除されたことをスタックのイベントステータスで確認します。

ステップ 2: Lake Formation クロスアカウント共有の前提条件

Lake Formation でリソースを共有する前に、タグベースのアクセスコントロール方式と名前付きリソース方式の両方に関する前提条件があります。

タグベースのアクセスコントロールのクロスアカウントデータ共有に関する前提条件を完了する

- クロスアカウントデータ共有要件の詳細については、「クロスアカウントデータ共有」という章の「前提条件」セクションを参照してください。

  データカタログリソースをクロスアカウントバージョン設定のバージョン 3 以降と共有するには、付与者は`AWSLakeFormationCrossAccountManager`アカウントの AWS 管理ポリシーで定義された IAM アクセス許可を持っている必要があります。

  [クロスアカウントバージョン設定] のバージョン 1 またはバージョン 2 を使用している場合は、タグベースのアクセスコントロール方式を使用してリソースへのクロスアカウントアクセス権を付与する前に、プロデューザーアカウントで以下の JSON 許可オブジェクトを Data Catalog リソースポリシーに追加する必要があります。これにより、`glue:EvaluatedByLakeFormationTags` が true であると、Data Catalog へのアクセス許可がコンシューマーアカウントに付与されます。また、この条件は、Lake Formation 許可タグを使用してリソースに対するアクセスをコンシューマーアカウントに許可した場合にも true になります。このポリシーは、アクセス許可を付与 AWS アカウント するすべての に必要です。

  次のポリシーは、`Statement` 要素内に配置する必要があります。IAM ポリシーの詳細については、次のセクションで説明します。

```
{
    "Effect": "Allow",
    "Action": [
        "glue:*"
    ],
    "Principal": {
        "AWS": [
            "consumer-account-id"
        ]
    },
    "Resource": [
        "arn:aws:glue:region:account-id:table/*",
        "arn:aws:glue:region:account-id:database/*",
        "arn:aws:glue:region:account-id:catalog"
```

```
    ],
    "Condition": {
        "Bool": {
            "glue:EvaluatedByLakeFormationTags": true
        }
    }
}
```

名前付きリソース方式のクロスアカウント共有に関する前提条件を完了する

1. アカウント内に Data Catalog リソースポリシーが存在しない場合、Lake Formation クロスアカウント付与を行うと、付与は通常どおり続行されます。一方、Data Catalog リソースポリシーが存在し、クロスアカウント付与に名前付きリソース方式を使用する場合、この付与を成功させるには、次のステートメントをポリシーに追加する必要があります。名前付きリソース方式またはタグベースのアクセスコントロール方式のいずれかのみを使用する場合は、このステップをスキップできます。このチュートリアルでは、両方の方式を評価するため、次のポリシーを追加する必要があります。

   次のポリシーは、Statement 要素内に配置する必要があります。IAM ポリシーの詳細については、次のセクションで説明します。

   ```
   {
             "Effect": "Allow",
             "Action": [
             "glue:ShareResource"
             ],
             "Principal": {
               "Service":"ram.amazonaws.com"
             },
             "Resource": [
                 "arn:aws:glue:region:account-id:table/*/*",
                 "arn:aws:glue:region:account-id:database/*",
                 "arn:aws:glue:region:account-id:catalog"
             ]
   }
   ```

2. 次に、 AWS Command Line Interface () を使用して AWS Glue Data Catalog リソースポリシーを追加しますAWS CLI。

タグベースのアクセスコントロール方式と名前付きリソース方式の両方を使用してクロスアカウント許可を付与する場合は、上記のポリシーを追加するときに EnableHybrid 引数を「true」に設定する必要があります。このオプションはコンソールでは現在サポートされていないため、glue:PutResourcePolicy APIと AWS CLIを使用する必要があります。

まず、ポリシードキュメント (policy.json など) を作成し、上記の 2 つのポリシーを追加します。*consumer-account-id* をグラント AWS アカウント を受け取る の***##### ID*** に、*region* をアクセス許可を付与するデータベースとテーブルを含むデータカタログのリージョンに、*account-id* をプロデューサー AWS アカウント ID に置き換えます。

次のコマンドを入力します AWS CLI 。*glue-resource-policy* は、正しい値 (file://policy.json など) に置き換えます。

```
aws glue put-resource-policy --policy-in-json glue-resource-policy --enable-hybrid
 TRUE
```

詳細については、「put-resource-policy」を参照してください。

ステップ 3: タグベースのアクセスコントロール方式を使用してクロスアカウント共有を実装する

このセクションでは、以下の大まかなステップについて説明します。

1. LF タグを定義する
2. LF タグをターゲットリソースに割り当てる
3. LF タグの許可をコンシューマーアカウントに付与する
4. データの許可をコンシューマーアカウントに付与する
5. (オプション) データベース、テーブル、列に対する許可を IAMAllowedPrincipals から取り消す
6. 共有テーブルへのリソースリンクを作成する
7. LF タグを作成してターゲットデータベースに割り当てる
8. LF タグのデータ許可をコンシューマーアカウントに付与する

LF タグを定義する

> **Note**
>
> プロデューサーアカウントにサインインしている場合は、サインアウトしてから以下のステップを開始してください。

1. https://console.aws.amazon.com/lakeformation/ でデータレイク管理者としてプロデューサーアカウントにサインインします。 CloudFormation のスタック作成時に指定したプロデューサーアカウント番号、IAM ユーザーネーム (デフォルトは `DatalakeAdminProducer`)、パスワードを使用します。
2. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) のナビゲーションペインで、[許可] の [管理ロールおよびタスク] を選択します。
3. [Add LF-Tag] (LF タグを追加) を選択します。

LF タグをターゲットリソースに割り当てる

LF タグをターゲットリソースに割り当て、データの許可を別のアカウントに付与する

データレイク管理者は、タグをリソースにアタッチできます。別のロールを使用する場合は、describe (記述) の許可と attach (アタッチ) の許可を別々のロールに付与する必要があります。

1. ナビゲーションペインの [Data Catalog] で、[Databases] (データベース) を選択します。
2. ターゲットデータベース (`lakeformation_tutorial_cross_account_database_tbac`) を選択し、[アクション] メニューの [LF タグの編集] を選択します。

   このチュートリアルでは、データベースに LF タグを割り当てますが、テーブルおよび列に LF タグを割り当てることもできます。
3. [Assign new LF-Tag] (新しい LF タグを割り当てる) を選択します。
4. 値が `public` の `Confidentiality` を追加します。
5. [Save] (保存) を選択します。

LF タグの許可をコンシューマーアカウントに付与する

プロデューサーアカウントで操作を続行し、LF タグへのアクセス許可をコンシューマーアカウントに付与します。

1. ナビゲーションペインで、[アクセス許可] の [LF タグとアクセス許可] を選択します。
2. [LF タグ] を選択し、コンシューマーアカウントと共有する LF タグのキーと値 (キーは `Confidentiality`、値は `public`) を選択します。
3. [Grant permissions] (アクセス許可の付与) を選択します。
4. [アクセス許可のタイプ] で、[LF タグのキーと値のペアのアクセス許可] を選択します。
5. [Principals] (プリンシパル) で、[External accounts] (外部アカウント) を選択します。
6. ターゲットの AWS アカウント ID を入力します。

   AWS アカウント 同じ組織内の が自動的に表示されます。それ以外の場合は、 AWS アカウント ID を手動で入力する必要があります。
7. [アクセス許可] で、[記述] を選択します。

   これはコンシューマーアカウントに付与されるアクセス許可です。付与可能な許可は、コンシューマーアカウントが他のプリンシパルに付与できる許可です。
8. [付与] を選択します。

   この時点でコンシューマーデータレイク管理者は、コンシューマーアカウントの Lake Formation コンソールで、共有されているポリシータグを確認できます ([アクセス許可] の [LF タグとアクセス許可] に移動します)。

データの許可をコンシューマーアカウントに付与する

ここで、データへのアクセス権をコンシューマーアカウントに付与します。そのためには、LF タグ式を指定し、この式に一致するテーブルまたはデータベースへのアクセス権をコンシューマーアカウントに付与します。

1. ナビゲーションペインで、[Permissions] (許可)、[Data lake permissions] (データレイクの許可) の順に移動し、[Grant] (付与) を選択します。
2. プリンシパル で、外部アカウントを選択し、ターゲット AWS アカウント ID を入力します。
3. [LF タグまたはカタログリソース] で、コンシューマーアカウントと共有されている [LF タグ] の [キー] および [値] ([キー] `Confidentiality` および [値] `public`) を選択します。
4. [許可] で、[LF タグに一致するリソース (推奨)] の [LF タグを追加] を選択します。
5. コンシューマーアカウントと共有するタグの キーおよび値 (キー `Confidentiality` および値 `public`) を選択します。
6. [Database permissions] (データベースの許可) で、[Database permissions] (データベースの許可) の [Describe] (記述) を選択して、データベースレベルでアクセス許可を付与します。

7. コンシューマーデータレイク管理者は、コンシューマーアカウントで共有しているポリシータグを、Lake Formation コンソールで確認できるはずです (https://console.aws.amazon.com/lakeformation/ で [許可]、[管理ロールおよびタスク]、[LF タグ] の順に移動します)。
8. [Grantable permissions] (付与可能な許可) で [Describe] (記述) を選択し、コンシューマーアカウントがそのユーザーに対してデータベースレベルの許可を付与できるようにします。
9. [Table and column permissions] (テーブルと列の許可) で [Select] (選択) を選択し、[Table permissions] (テーブルの許可) の [Describe] (記述) を選択します。
10. [Select] (選択) を選択し、[Grantable permissions] (付与可能な許可) で [Describe] (記述) を選択します。
11. [付与] を選択します。

(オプション) データベース、テーブル、列に対する許可を **IAMAllowedPrincipals** から取り消す

このチュートリアルの最初に、Lake Formation の Data Catalog 設定を変更しました。その部分をスキップした場合は、このステップが必要です。Lake Formation の Data Catalog 設定を変更している場合は、このステップをスキップできます。

このステップでは、データベースまたはテーブルに対するデフォルトの Super 許可を `IAMAllowedPrincipals` から取り消す必要があります。詳細については、「ステップ 4: データストアを Lake Formation 許可モデルに切り替える」を参照してください。

`IAMAllowedPrincipals` の許可を取り消す前に、Lake Formation で既存の IAM プリンシパルに必要な許可を付与していることを確認します。これには、以下の 3 つのステップを使用します。

1. Lake Formation の `GetDataAccess` アクションを使用して IAM 許可をターゲットの IAM ユーザーまたはロールに追加します (IAM ポリシーを使用)。
2. ターゲットの IAM ユーザーまたはロールに対して Lake Formation データの許可 (変更、選択など) を付与します。
3. 次に、`IAMAllowedPrincipals` の許可を取り消します。上記のステップに従わない場合、`IAMAllowedPrincipals` の許可を取り消した後では、既存の IAM プリンシパルからターゲットデータベースまたは Data Catalog にアクセスできなくなる可能性があります。

   `IAMAllowedPrincipals` の Super 許可の取り消すが必要になるのは、Lake Formation 許可モデルを (IAM ポリシーモデルの代わりに) 適用して、単一のアカウント内または複数のアカウント間でユーザーアクセスを管理する場合です。従来の IAM ポリシーモデルを維持する他のテーブルの場合。`IAMAllowedPrincipals` の許可を取り消す必要はありません。

この時点で、コンシューマーアカウントのデータレイク管理者は、コンシューマーアカウントで共有しているデータベースとテーブルを Lake Formation コンソールで確認できるはずです (https://console.aws.amazon.com/lakeformation/ で [Data Catalog] (データカタログ)、[databases] (データベース) の順に移動します)。確認できない場合は、以下が適切に設定されているかどうかをチェックします。

1. 正しいポリシータグおよび値がターゲットデータベースおよびテーブルに割り当てられている。
2. 正しいタグの許可およびデータの許可がコンシューマーアカウントに割り当てられている。
3. データベースまたはテーブルに対するデフォルトの Super 許可を `IAMAllowedPrincipals` から取り消している。

共有テーブルへのリソースリンクを作成する

リソースをアカウント間で共有すると、共有リソースはコンシューマーアカウントの Data Catalog に配置されません。これらをアクセス可能にして、共有テーブルの基になるデータに対して Athena などのサービスでクエリを実行するには、共有テーブルへのリソースリンクを作成する必要があります。リソースリンクは、ローカルまたは共有のデータベースやテーブルへのリンクである Data Catalog オブジェクトです。詳細については、「リソースリンクの作成」を参照してください。リソースリンクを作成することで、以下のことができます。

- Data Catalog のリソース命名ポリシーに適合した別の名前をデータベースまたはテーブルに割り当てる。
- Athena や Redshift Spectrum などのサービスを使用して、共有データベースやテーブルに対してクエリを実行する。

リソースリンクを作成するには、以下のステップを実行します。

1. コンシューマーアカウントにサインインしている場合は、サインアウトします。
2. コンシューマーアカウントのデータレイク管理者としてサインインします。 CloudFormation スタックの作成時に指定したコンシューマーアカウント ID、IAM ユーザー名 (デフォルト DatalakeAdminConsumer)、パスワードを使用します。
3. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) のナビゲーションペインで、[Data Catalog]、[Databases] (データベース)の順に移動し、共有データベース `lakeformation_tutorial_cross_account_database_tbac` を選択します。

データベースが表示されない場合は、上記の手順に戻り、すべてが正しく設定されているかどうかを確認します。

4. [View details] (詳細の表示) を選択します。
5. 共有テーブル amazon_reviews_table_tbac を選択します。
6. [Actions] (アクション) メニューで、[Create resource link] (リソースリンクの作成) を選択します。
7. [Resource link name] (リソースリンク名) に名前 (このチュートリアルでは amazon_reviews_table_tbac_resource_link) を入力します。
8. Database で、リソースリンクが作成されるデータベースを選択します (この投稿では、n CloudFormationスタックによってデータベース が作成されましたlakeformation_tutorial_cross_account_database_consumer)。
9. [作成] を選択します。

   リソースリンクが [Data Catalog] の [Tables] (テーブル) の下に表示されます。

LF タグを作成してターゲットデータベースに割り当てる

Lake Formation のタグは、リソースと同じ Data Catalog 内に存在します。つまり、プロデューサーアカウントで作成したタグは、コンシューマーアカウントでリソースリンクへのアクセスを許可しても利用できません。コンシューマーアカウントでリソースリンクを共有する場合、LF タグベースのアクセスコントロールを使用するには、コンシューマアカウントで別個の LF タグのセットを作成する必要があります。

1. コンシューマアカウントで LF タグを定義します。このチュートリアルでは、キーとして Division を使用し、値として sales、marketing、analyst を使用します。
2. LF タグのキー Division および値 analyst を、リソースリンクを作成したデータベース lakeformation_tutorial_cross_account_database_consumer に割り当てます。

LF タグのデータ許可をコンシューマーに付与する

最後のステップとして、LF タグのデータ許可をコンシューマーに付与します。

1. ナビゲーションペインで、[Permissions] (許可)、[Data lake permissions] (データレイクの許可) の順に移動し、[Grant](付与) を選択します。

2. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール)を選択し、ユーザー `DataAnalyst` を選択します。
3. [LF タグまたはカタログリソース] で、[LF タグに一致するリソース] (推奨) を選択します。
4. キーとして Divison、値として analyst を選択します。
5. [Database permissions] (データベースの許可) で、[Database permissions] (データベースの許可) の [Describe] (記述) を選択します。
6. [Table and column permissions] (テーブルと列の許可) で、[Select] (選択) を選択し、[Table permissions] (テーブルの許可) の [Describe] (記述) を選択します。
7. [付与] を選択します。
8. ユーザー `DataAnalyst` に対してこれらのステップを繰り返します。ここで、LF タグのキーは `Confidentiality`、値は `public` です。

   この時点で、コンシューマアカウントのデータアナリストユーザーは、データベースとリソースリンクを見つけて、Athena コンソール (https://console.aws.amazon.com/athena/) を介して共有テーブルにクエリを実行できるはずです。見つからない場合は、以下が適切に設定されているかどうかを確認します。

   - 共有テーブルへのリソースリンクが作成されている
   - プロデューサーアカウントが共有する LF タグへのアクセスをユーザーに許可している
   - リソースリンク、およびリソースリンクを作成したデータベースに関連付けられた LF タグへのアクセスをユーザーに許可している
   - リソースリンク、およびリソースリンクを作成したデータベースに正しい LF タグが割り当てられているかどうかを確認する

ステップ 4: 名前付きリソース方式を実装する

名前付きリソース方式を使用するには、以下の大まかなステップに従います。

1. (オプション) データベース、テーブル、列に対する許可を `IAMAllowedPrincipals` から取り消す
2. データの許可をコンシューマーアカウントに付与する
3. リソース共有を受け入れます AWS Resource Access Manager。
4. 共有テーブルへのリソースリンクを作成する
5. 共有テーブルへのデータ許可をコンシューマーに付与する

6. リソースリンクへのデータ許可をコンシューマーに付与する

(オプション) データベース、テーブル、列に対する許可を **IAMAllowedPrincipals** から取り消す

- このチュートリアルの最初で、Lake Formation の Data Catalog 設定を変更しました。その部分をスキップした場合は、このステップが必要です。手順については、前のセクションのオプションのステップを参照してください。

データの許可をコンシューマーアカウントに付与する

1.

> **Note**
>
> プロデューサーアカウントに別のユーザーとしてサインインしている場合は、まずサインアウトします。

   CloudFormation スタックの作成時に指定された AWS アカウント ID、IAM ユーザー名 (デフォルトは `DatalakeAdminProducer`)、パスワードを使用して、プロデューサーアカウントのデータレイク管理者を使用して、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールにサインインします。
2. [Permissions] (許可) ページの [Data lake Permissions] (データレイクの許可) で、[Grant] (付与) を選択します。
3. プリンシパルで外部アカウントを選択し、1 つ以上の AWS アカウント IDsまたは AWS 組織 IDsを入力します。詳細については、「AWS Organizations」を参照してください。

   プロデューサーアカウントが属し、同じ組織 AWS アカウント 内の組織が自動的に表示されます。表示されない場合は、アカウント ID または組織 ID を手動で入力します。
4. [LF タグまたはカタログリソース] で、`Named data catalog resources` を選択します。
5. [Databases] (データベース) で、データベース `lakeformation_tutorial_cross_account_database_named_resource` を選択します。
6. [Add LF-Tag] (LF タグを追加) を選択します。
7. [Tables] (テーブル) で、[All tables] (すべてのテーブル) を選択します。
8. [Table column permissions] (テーブル列の許可) で、[Select] (選択) を選択し、[Table permissions] (テーブルの許可) の [Describe] (記述) を選択します。

9. [Select] (選択) を選択し、[Grantable permissions] (付与可能な許可) の [Describe] (記述) を選択します。
10. (オプション) [Data permissions] (データの許可) で、列レベルの許可の管理が必要な場合は、[Simple column-based access] (シンプルな列ベースのアクセス) を選択します。
11. [付与] を選択します。

`IAMAllowedPrincipals` の許可を取り消していない場合は、Grant permissions (許可の付与) 失敗エラーが表示されます。この時点で、ターゲットテーブルが 経由でコンシューマーアカウント AWS RAM と Permissions, Data permissions で共有されていることがわかります。

からリソース共有を受け入れる AWS RAM

> **Note**
>
> このステップは、組織 AWS アカウントベースの共有ではなく、ベースの共有にのみ必要です。

1. CloudFormation スタックの作成時に指定された IAM ユーザー名 (デフォルトは DatalakeAdminConsumer) とパスワードを使用して、コンシューマーアカウントデータレイク管理者を使用して https://console.aws.amazon.com/connect/ で AWS コンソールにサインインします。
2. AWS RAM コンソールのナビゲーションペインの「自分と共有」の「リソース共有」で、共有された Lake Formation リソースを選択します。[Status] (ステータス) は [Pending] (保留) になっているはずです。
3. [Actions] (アクション)、[Grant] (付与) の順に選択します。
4. リソースの詳細を確認し、[Accept resource share] (リソース共有を承認) を選択します。

   この時点で、コンシューマーアカウントのデータレイク管理者は、共有しているリソースを Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) で確認できるはずです ([Data Catalog] (データカタログ)、[Databases] (データベース) の順に移動します)。

共有テーブルへのリソースリンクを作成する

- 「ステップ 3: タグベースのアクセスコントロール方式を使用してクロスアカウント共有を実装する」 (ステップ 6) の手順に従って、共

有テーブルへのリソースリンクを作成します。リソースリンク名を `amazon_reviews_table_named_resource_resource_link` とします。リソースリンクをデータベース `lakeformation_tutorial_cross_account_database_consumer` 内に作成します。

共有テーブルへのデータ許可をコンシューマーに付与する

共有テーブルへのデータ許可をコンシューマーに付与するには、以下のステップを実行します。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) で、[Permissions] (許可)、[Data lake permissions] (データレイクの許可) の順に移動し、[Grant] (付与) を選択します。
2. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール)を選択し、ユーザー `DataAnalyst` を選択します。
3. [LF タグまたはカタログリソース] で、[名前付きの Data Catalog リソース] を選択します。
4. [Databases] (データベース) で、データベース `lakeformation_tutorial_cross_account_database_named_resource` を選択します。データベースがドロップダウンリストに表示されない場合は、[Load more] (さらにロード) を選択します。
5. [Tables] (テーブル) で、テーブル `amazon_reviews_table_named_resource` を選択します。
6. [Table and column permissions] (テーブルと列の許可) で、[Select] (選択) を選択し、[Table permissions] (テーブルの許可) の [Describe] (記述) を選択します。
7. [付与] を選択します。

リソースリンクへのデータ許可をコンシューマーに付与する

データレイクユーザーに対しては、共有テーブルへのアクセス許可だけでなく、リソースリンクへのアクセス許可も付与する必要があります。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) で、[Permissions] (許可)、[Data lake permissions] (データレイクの許可) の順に移動し、[Grant] (付与) を選択します。
2. [Principals] (プリンシパル) で、[IAM users and roles] (IAM ユーザーおよびロール)を選択し、ユーザー `DataAnalyst` を選択します。

3. [LF タグまたはカタログリソース] で、[名前付きの Data Catalog リソース] を選択します。
4. [Databases] (データベース) で、データベース `lakeformation_tutorial_cross_account_database_consumer` を選択します。データベースがドロップダウンリストに表示されない場合は、[Load more] (さらにロード) を選択します。
5. [Tables] (テーブル) で、テーブル `amazon_reviews_table_named_resource_resource_link` を選択します。
6. [Resource link permissions] (リソースリンクの許可) で、[Resource link permissions] (リソースリンクの許可) の [Describe] (記述) を選択します。
7. [付与] を選択します。

   この時点で、コンシューマアカウントのデータアナリストユーザーは、データベースとリソースリンクを見つけて、Athena コンソールを介して共有テーブルにクエリを実行できるはずです。

   見つからない場合は、以下が適切に設定されているかどうかを確認します。

   - 共有テーブルへのリソースリンクが作成されている
   - プロデューサーアカウントが共有するテーブルへのアクセスをユーザーに許可している
   - リソースリンク、およびリソースリンクを作成したデータベースへのアクセスをユーザーに許可している

ステップ 5: AWS リソースをクリーンアップする

に不要な料金が発生しないように AWS アカウント、このチュートリアルで使用した AWS リソースを削除できます。

1. プロデューサーアカウントを使用して Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインし、以下を削除または変更します。

   - AWS Resource Access Manager リソース共有
   - Lake Formation タグ
   - CloudFormation スタック
   - Lake Formation 設定
   - AWS Glue Data Catalog
2. コンシューマーアカウントを使用して Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインし、以下を削除または変更します。

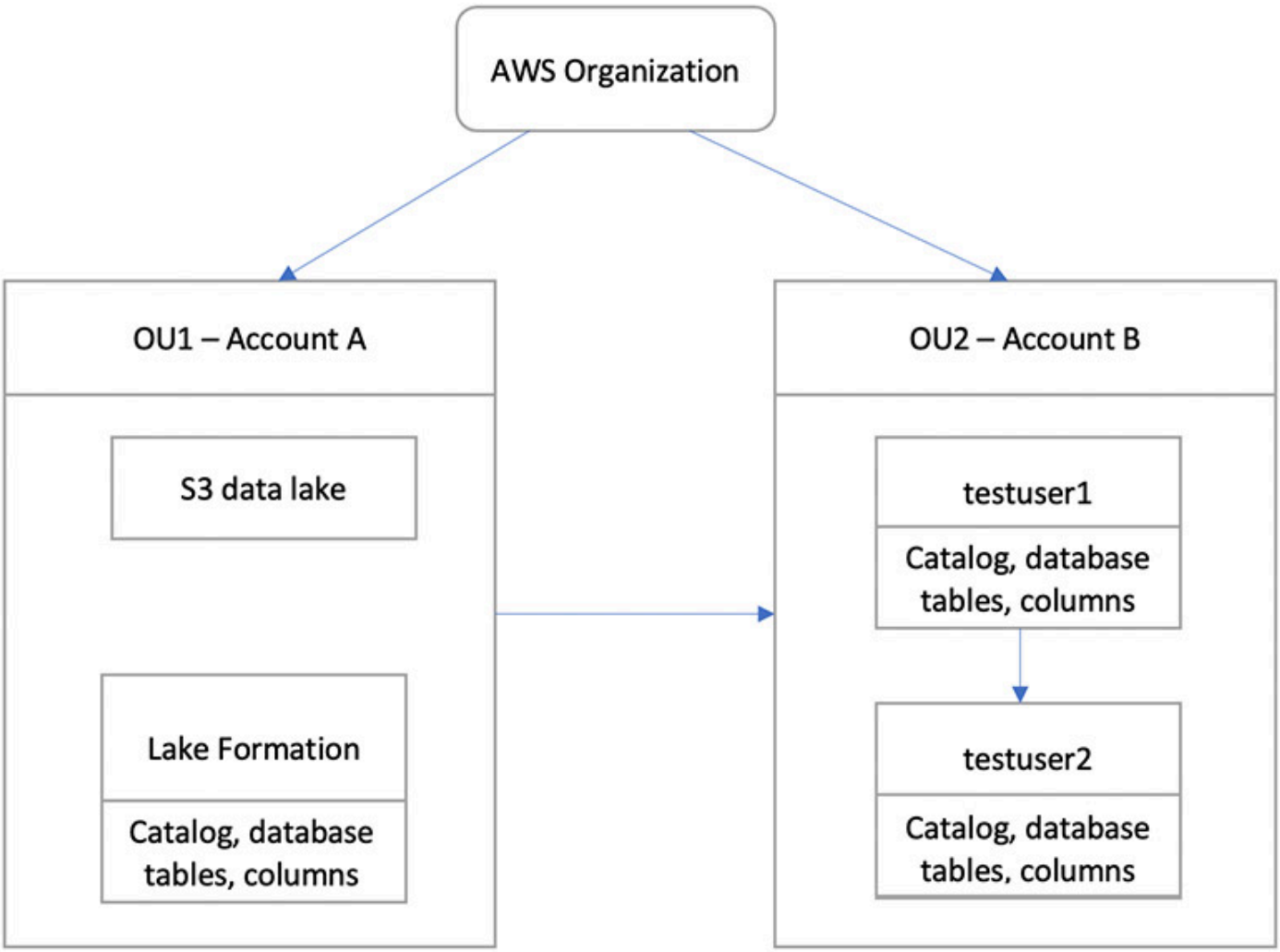
- Lake Formation タグ
- CloudFormation スタック

Lake Formation のきめ細かなアクセスコントロールを使用したデータレイクの共有

このチュートリアルでは、複数の を管理するときに Lake Formation を使用してデータセットをすばやく簡単に共有する方法をstep-by-stepで説明します AWS アカウント AWS Organizations。機密データへのアクセスを制御するには、きめ細かな許可を定義します。

次の手順では、アカウント A のデータレイク管理者がアカウント B に対してきめ細かなアクセスを付与する方法と、アカウント B のユーザーがデータスチュワードとしてアカウント内の他のユーザーに対して共有テーブルへのきめ細かなアクセスを許可する方法も示します。各アカウント内のデータスチュワードは、各自が管理するユーザーに対して独自にアクセス権を委任し、各チームや基幹業務 (LOB) に自治権を付与できます。

ユースケースでは、 を使用して を管理していること AWS Organizations を前提としています AWS アカウント。1 つの組織単位 (OU1) のアカウント A のユーザーは、OU2 のアカウント B のユーザーに対してアクセス権を付与します。Organizations を使用していない場合 (少数のアカウントしか持っていない場合など) でも、同じアプローチを使用できます。次の図は、データレイク内でのデータセットに対するきめ細かなアクセスコントロールを示しています。データレイクは、アカウント A にあります。アカウント A のデータレイク管理者は、アカウント B に対してきめ細かなアクセス権を提供しています。この図は、アカウント B のユーザーがアカウント A のデータレイクテーブルの列レベルのアクセス権をアカウント B の別のユーザーに提供していることも示しています。



トピック

- 対象者
- 前提条件
- ステップ 1: 別のアカウントに対してきめ細かなアクセスを提供する
- ステップ 2: 同じアカウント内のユーザーにきめ細かなアクセスを提供する

対象者

このチュートリアルは、データスチュワード、データエンジニア、データアナリストを対象としています。次の表は、このチュートリアルで使用するロールのリストです。

ロール	説明
IAM 管理者	AWS 管理ポリシーを持つユーザー: `AdministratorAccess` 。
データレイク管理者	AWS 管理ポリシーを持つユーザー: ロールに`AWSLakeFormationDataAdmin` アタッチされています。
データアナリスト	AWS 管理ポリシー: が`AmazonAthenaFullAccess` アタッチされているユーザー。

前提条件

このチュートリアルを開始する前に、適切なアクセス許可を持つ管理ユーザーとしてサインインするために AWS アカウント 使用できる が必要です。詳細については、「初期設定 AWS タスクを完了する」を参照してください。

このチュートリアルでは、ユーザーが IAM に精通していることを前提としています。IAM については、「IAM ユーザーガイド」を参照してください。

このチュートリアルでは、以下のリソースが必要です。

- 2 つの組織単位
  - OU1 — アカウント A を含む
  - OU2 — アカウント B を含む
- アカウント A の Amazon S3 データレイクの口ケーション (バケット)
- アカウント A のデータレイク管理者ユーザー。データレイク管理者は、Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) または Lake Formation API の `PutDataLakeSettings` 操作を使用して作成できます。
- アカウント A に設定した Lake Formation と、アカウント A の Lake Formation に登録した Amazon S3 データレイクの口ケーション。
- 次の IAM マネージドポリシーを持つ、アカウント B の 2 人のユーザー。
  - testuser1 – 管理 AWS ポリシーが`AWSLakeFormationDataAdmin`アタッチされています。
  - testuser2 – AWS 管理ポリシーが`AmazonAthenaFullAccess`アタッチされています。

- アカウント B の Lake Formation データベース内のデータベース testdb。

ステップ 1: 別のアカウントに対してきめ細かなアクセスを提供する

アカウント A のデータレイク管理者がアカウント B に対してきめ細かなアクセスを提供する方法について学習します。

別のアカウントに対してきめ細かなアクセスを許可する

1. データレイク管理者としてアカウント A の https://console.aws.amazon.com/connect/ で にサインイン AWS マネジメントコンソール します。
2. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開き、[Get started] (今すぐ始める) を選択します。
3. ナビゲーションペインで、[DataDatabases] (データベース) を選択します。
4. [Create database] (データベースを作成) を選択します。
5. [Database] (データベース) の詳細セクションで、[Database] (データベース) を選択します。
6. [Name] (名前) に名前を入力します (このチュートリアルでは `sampledb01` を使用します)。
7. [Use only IAM access control for new tables in this database] (このデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) が選択されていることを確認します。これが選択されていないと、Lake Formation からアクセスをコントロールできます。
8. [データベースの作成] を選択します。
9. [Database] (データベース) ページで、データベース `sampledb01` を選択します。
10. [Actions] (アクション) メニューで、[Grant] (付与) を選択します。
11. [Grant permissions] (許可の付与) セクションで、[External account (外部アカウント) を選択します。
12. AWS アカウント ID または AWS 組織 ID の場合は、OU2 でアカウント B のアカウント ID を入力します。
13. [Table] (テーブル) で、アカウント B にアクセスを許可するテーブルを選択します (このチュートリアルでは、テーブル `acc_a_area` を使用します)。オプションとして、テーブル内の列へのアクセスを許可することもできます (このチュートリアルでは、これを行います)。
14. [Include columns] (列を含める) で、アカウント B にアクセスを許可する列を選択します (このチュートリアルでは、タイプ、名前、識別子への許可を付与します)。
15. [Columns] (列) で、[Include columns] (列を含める) を選択します。

16. [Table permissions] (テーブルの許可) で、[Select] (選択) を選択します。
17. [Grantable permissions] (付与可能な許可) で、[Select] (選択) を選択します。付与可能な許可を設定することで、アカウント B の管理者ユーザーはアカウント B の他のユーザーに許可を付与できるようになります。
18. [付与] を選択します。
19. ナビゲーションペインで、[Table] (テーブル) を選択します。
20. アクセス権限を持つ AWS アカウント および AWS 組織セクションに 1 つのアクティブな接続が表示されます。

リソースリンクを作成する

Amazon Athena などの統合サービスは、複数のアカウントをまたいでデータベースやテーブルに直接アクセスできません。したがって、リソースリンクを作成する必要があります。Athena がアカウント内のリソースリンクを使用して、他のアカウントのデータベースやテーブルにアクセスできるようにします。テーブル (`acc_a_area`) へのリソースリンクを作成し、アカウント B のユーザーが Athena を使用してデータをクエリできるようにします。

1. としてアカウント B の https://console.aws.amazon.com/connect/ で AWS コンソールにサインインしますtestuser1。
2. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) のナビゲーションペインで [Tables] (テーブル) を選択します。アカウント A がアクセスを提供しているテーブルが表示されます。
3. テーブル `acc_a_area` を選択します。
4. [Actions] (アクション) メニューで、[Create resource link] (リソースリンクを作成) を選択します。
5. [Resource link name] (リソースリンク名) に名前 (このチュートリアルでは `acc_a_area_rl`) を入力します。
6. [Database] (データベース) で、データベース (`testdb`) を選択します。
7. [Create] (作成) を選択します。
8. ナビゲーションペインで、[Tables] (テーブル) を選択します。
9. テーブル `acc_b_area_rl` を選択します。
10. [Actions] (アクション) メニューで、[View data] (データの表示) を選択します。

    Athena コンソールにリダイレクトされ、データベースとテーブルが表示されます。

これで、テーブルに対してクエリを実行し、アカウント B から testuser1 にアクセスを許可した先の列値を確認できます。

ステップ 2: 同じアカウント内のユーザーにきめ細かなアクセスを提供する

このセクションでは、アカウント B のユーザー (testuser1) がデータスチュワードとして同じアカウント内の別のユーザー (testuser2) に対し、共有テーブル aac_b_area_rl 内の列名へのきめ細かなアクセスを提供する方法を示します。

同じアカウント内のユーザーに対してきめ細かなアクセスを許可する

1. としてアカウント B の https://console.aws.amazon.com/connect/ で AWS コンソールにサインインしますtestuser1。
2. Lake Formation コンソールのナビゲーションペインで、[Tables] (テーブル) を選択します。

   テーブルに対する許可は、リソースリンクを使用して付与できます。これを実行するには、[Tables] (テーブル) ページでリソースリンク acc_b_area_rl を選択し、[Actions] (アクション) メニューで、[Grant on target] (ターゲットに対して付与) を選択します。
3. [Grant permissions] (許可の付与) セクションで、[My account] (マイアカウント) を選択します。
4. [IAM users and roles] (IAM ユーザーおよびロール) で、ユーザー testuser2 を選択します。
5. [Column] (列) で、列名を選択します。
6. [Table permissions] (テーブルの許可) で、[Select] (選択) を選択します。
7. [付与] を選択します。

   リソースリンクの作成後は、作成したユーザーのみがそのリンクを表示してアクセスできます。アカウント内の他のユーザーにリソースリンクへのアクセスを許可するには、リソースリンク自体に対する許可を付与する必要があります。DESCRIBE 許可または DROP 許可を付与する必要があります。[Tables] (テーブル) ページでテーブルを再び選択し、[Actions] (アクション) メニューで [Grant] (付与) を選択します。
8. [Grant permissions] (許可の付与) セクションで、[My account] (マイアカウント) を選択します。
9. [IAM users and roles] (IAM ユーザーおよびロール) で、ユーザー testuser2 を選択します。
10. [Resource link permissions] (リソースリンクの許可) で、[Describe] (記述) を選択します。
11. [付与] を選択します。
12. アカウント B の AWS コンソールに としてサインインしますtestuser2。

Athena コンソール (https://console.aws.amazon.com/athena/) に、データベースとテーブル `acc_b_area_rl` が表示されます。これで、テーブルに対してクエリを実行し、`testuser2` からアクセス可能となった列値を確認できます。

Lake Formation 許可へのオンボーディング

AWS Lake Formation は AWS Glue Data Catalog (データカタログ) を使用して、Amazon S3 データレイクのメタデータと、Amazon Redshift などの外部データソースをカタログ、データベース、テーブルの形式で保存します。データカタログ内のメタデータは、カタログ、データベース、テーブルで構成される 3 レベルのデータ階層に分類されます。さまざまなソースからのデータをカタログと呼ばれる論理コンテナにまとめます。データベースはテーブルのコレクションです。Data Catalog には、リソースリンクも含まれています。これは、外部アカウントの共有データベースとテーブルへのリンクで、データレイク内のデータへのクロスアカウントアクセスに使用されます。各 AWS アカウントには、 AWS リージョンごとに 1 つのデータカタログがあります。

Lake Formation には、Amazon S3 内の基盤となるデータを含むデータカタログのカタログ、データベース、テーブル、列へのアクセスを許可または取り消すためのリレーショナルデータベース管理システム (RDBMS) のアクセス許可モデルが用意されています。

Lake Formation 許可モデルの詳細について学ぶ前に、以下の背景情報を確認しておくことが役に立ちます。

- Lake Formation によって管理されるデータレイクは、Amazon Simple Storage Service (Amazon S3) 内の指定された口ケーションに置かれます。データカタログにはカタログオブジェクトも含まれています。各カタログは、Amazon Redshift データウェアハウス、 Amazon DynamoDB データベース、および Snowflake、MySQL の他 30 を超える外部データソースを含むサードパーティーデータソースソースなどのソースからのデータを表し、これらはフェデレーティッドコネクタを介して統合されています。
- Lake Formation は、データレイクにインポートされるログやリレーショナルデータベース内のデータなどのソースデータ、および Amazon S3 内のデータレイクにあるデータに関するメタデータが含まれた Data Catalog を維持します。データカタログには、Amazon S3 以外の外部データソースからのデータに関するメタデータも含まれています。メタデータは、カタログ、データベース、テーブルとして編成されます。メタデータテーブルには、スキーマ、ロケーション、パーティショニング、およびそれらが表すデータに関するその他の情報が含まれています。メタデータデータベースは、テーブルのコレクションです。
- Lake Formation Data Catalog は、AWS Glue が使用する Data Catalog と同じです。AWS Glue クローラを使用して Data Catalog テーブルを作成し、AWS Glue 抽出、変換、ロード (ETL) ジョブを使用してデータレイク内の基盤となるデータを投入することができます。
- データカタログ内のカタログ、データベース、テーブルは、データカタログリソースと呼ばれます。データカタログ内のテーブルは、Amazon S3 のデータソースまたは表形式データ内のテー

ブルと区別するために、メタデータテーブルと呼ばれます。メタデータテーブルがポイントする Amazon S3 またはデータソース内のデータは、基盤となるデータと呼ばれます。

- プリンシパルは、ユーザーまたはロール、Amazon Quick ユーザーまたはグループ、SAML プロバイダーを介して Lake Formation で認証するユーザーまたはグループ、またはクロスアカウントアクセスコントロール、 AWS アカウント ID、組織 ID、または組織単位 ID です。
- AWS Glue クローラはメタデータテーブルを作成しますが、Lake Formation コンソール、API、または AWS Command Line Interface () を使用してメタデータテーブルを手動で作成することもできますAWS CLI。メタデータテーブルを作成するときは、ロケーションを指定する必要があります。データベースを作成するときは、ロケーションはオプションです。テーブルロケーションは、Amazon S3 ロケーション、または Amazon Relational Database Service (Amazon RDS) データベースなどのデータソースロケーションにすることができます。データベースロケーションは、常に Amazon S3 ロケーションです。
- Amazon Athena および Amazon Redshift などの Lake Formation と統合するサービスは、メタデータの取得、またはクエリを実行するための認可の確認を実行するために Data Catalog にアクセスできます。統合されたサービスの完全なリストについては、「AWS Lake Formation とのサービス統合」を参照してください。

トピック

- Lake Formation 許可の概要
- Lake Formation のペルソナと IAM 許可のリファレンス
- データレイクのデフォルト設定の変更
- 黙示的な Lake Formation 許可
- Lake Formation 許可のリファレンス
- IAM アイデンティティセンターの統合
- データレイクへの Amazon S3 ロケーションの追加
- ハイブリッドアクセスモード
- でのオブジェクトの作成 AWS Glue Data Catalog
- Lake Formation でのワークフローを使用したデータのインポート

Lake Formation 許可の概要

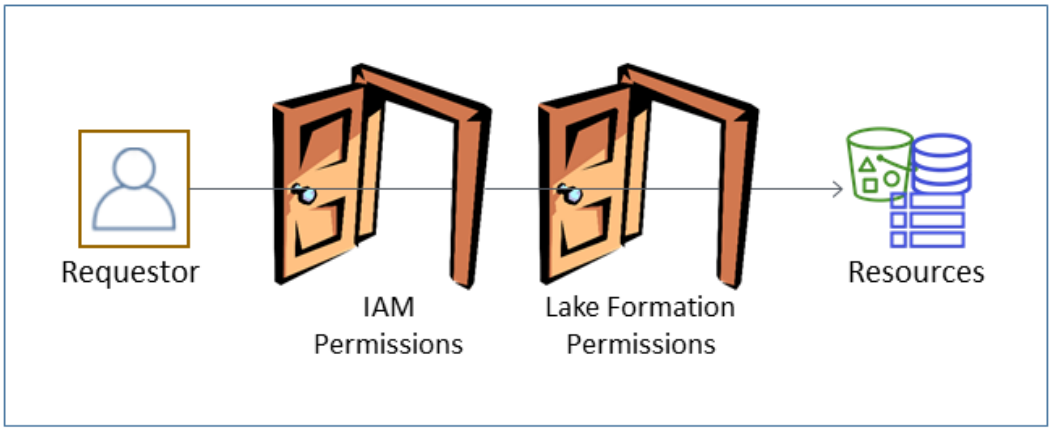
AWS Lake Formationには、2 つの主な許可タイプがあります。

- メタデータアクセス – データカタログリソースに対する許可 (データカタログ許可)。

  これらの許可は、プリンシパルが Data Catalog 内のメタデータデータベースとテーブルの作成、読み取り、更新、および削除を実行できるようにします。
- 基盤となるデータアクセス – Amazon Simple Storage Service (Amazon S3) 内の口ケーションに対するアクセス許可 (データアクセス許可とデータロケーション許可)。
  - データレイクのアクセス許可により、プリンシパルが基盤となる Amazon S3 口ケーション (データカタログリソースがポイントするデータ) に対するデータの読み取りと書き込みが実行できるようになります。
  - データロケーション許可は、プリンシパルが特定の Amazon S3 口ケーションをポイントするメタデータデータベースとテーブルの作成と変更を実行できるようにします。

どちらの領域でも、Lake Formation は Lake Formation アクセス許可と AWS Identity and Access Management (IAM) アクセス許可の組み合わせを使用します。IAM 許可モデルは、IAM ポリシーで構成されます。Lake Formation 許可モデルは、`Grant SELECT on` *`tableName`* `to` *`userName`* のような、DBMS 形式の GRANT/REVOKE コマンドとして実装されます。

プリンシパルが Data Catalog リソース、または基盤となるデータへのアクセスをリクエストするときにリクエストが成功するには、そのリクエストが IAM と Lake Formation の両方による許可チェックに合格する必要があります。



Lake Formation 許可は Data Catalog リソース、Amazon S3 口ケーション、およびこれらの口ケーションにある基盤となるデータへのアクセスを制御します。IAM 許可は、Lake Formation、および AWS Glue の API とリソースへのアクセスを制御します。このため、Data Catalog にメタデータテーブルを作成するための Lake Formation 許可 (`CREATE_TABLE`) を持っていても、`glue:CreateTable` API に対する IAM の許可を持っていなければ、操作が失敗します。(`glue:` 許可である理由は、Lake Formation が AWS Glue Data Catalog を使用するからです。)

> **Note**
>
> Lake Formation 許可は、それらが付与されたリージョンのみで適用されます。

AWS Lake Formation では、各プリンシパル (ユーザーまたはロール) に Lake Formation が管理するリソースでアクションを実行する権限が必要です。プリンシパルは、データレイク管理者、または Lake Formation 許可を付与する許可を持つ別のプリンシパルから必要な認可を付与されます。

Lake Formation 許可をプリンシパルに付与するときは、その許可を別のプリンシパルに渡す能力をオプションで付与できます。

Lake Formation API、 AWS Command Line Interface (AWS CLI)、または Lake Formation コンソールのデータアクセス許可とデータロケーションページを使用して、Lake Formation アクセス許可を付与および取り消すことができます。

細粒度のアクセスコントロールのための方式

データレイクでは、データに対する細粒度のアクセスコントロールを持つことが目標になります。これは、Lake Formation では Data Catalog リソースと Amazon S3 ロケーションに対する細粒度のアクセスコントロールを意味します。細粒度のアクセスコントロールは、以下の方式のいずれかを使用して達成することができます。

方式	Lake Formation 許可	IAM 許可	コメント
方式 1	オープン	細粒度	AWS Glue との後方互換性のためのデフォルト方式です。 • オープンとは、特別なアクセス許可である Super がグループ IAMAllowedPrincipals に付与されていることを意味し、この場合、IAMAllowedPrincipals が自動的に作成され、IAM ポリシーによってデータカタログリソースへのアクセスが許可されているすべての IAM ユーザーとロールが含まれます。Super アクセス許可は、その

方式	Lake Formation 許可	IAM 許可	コメント
			許可が付与されるデータベースやテーブルに対して、サポートされているすべての Lake Formation オペレーションをプリンシパルが実行できるようにします。これによって、Data Catalog リソースと Amazon S3 ロケーションへのアクセスは、実質的に IAM ポリシーのみで制御されることになります。詳細については、「データレイクのデフォルト設定の変更」および「AWS Lake Formation モデルへのAWS Glueデータアクセス許可のアップグレード」を参照してください。 • きめ細かなとは、IAM ポリシーがデータカタログリソースおよび個々の Amazon S3 バケットに対するすべてのアクセスを制御することを意味します。 Lake Formation コンソールでは、この方式が [Use only IAM access control] (IAM アクセスコントロールのみを使用する) として表示されます。

方式	Lake Formation 許可	IAM 許可	コメント
方式 2	細粒度	粗粒度	これは、推奨される方法です。 • きめ細かなアクセスとは、データカタログリソース、Amazon S3 の場所、およびこれらの場所にある基盤となるデータに対する限定的な Lake Formation アクセス許可を個々のプリンシパルに付与することを意味します。 • 大まかなとは、個々のオペレーションや Amazon S3 の場所へのアクセスに対するアクセス許可が広範囲であることを意味します。例えば、粗粒度の IAM ポリシーには、`"glue:CreateTables"` ではなく `"glue:*"` または `"glue:Create*"` が含まれているため、プリンシパルがカタログオブジェクトを作成できるかどうかは Lake Formation 許可で制御することになります。また、プリンシパルが作業を実行するために必要な API へのアクセス権をプリンシパルに提供しても、他の API とリソースはロックダウンするという意味でもあります。例えば、プリンシパルが Data Catalog リソースを作成し、ワークフローを作成して実行することはできても、AWS Glue 接続やユーザー定義の関数を作成することはできないという IAM ポリシーを作成するなどがあります。このセクションで後述の例を参照してください。

Important

以下の点に注意してください。

- Lake Formation では、既存の AWS Glue Data Catalog 動作との互換性のために、[Use only IAM access control] (IAM アクセスコントロールのみを使用する) がデフォルトで有効になっています。これらの設定は、Lake Formation 許可の使用への移行後に無効化することをお勧めします。詳細については、「データレイクのデフォルト設定の変更」を参照してください。
- データレイク管理者とデータベース作成者には、理解しておく必要がある黙示的な Lake Formation 許可があります。詳細については、「黙示的な Lake Formation 許可」を参照してください。

メタデータのアクセスコントロール

Data Catalog リソースのアクセスコントロールに関する以下の説明は、Lake Formation 許可を使用した細粒度のアクセスコントロールと、IAM ポリシーを使用した粗粒度のアクセスコントロールを前提としています。

Data Catalog リソースに対する Lake Formation 許可を付与するには、以下の 2 つの異なる方式があります。

- 名前付きリソースでのアクセスコントロール – この方式では、データベース名またはテーブル名を指定することで、特定のデータベースまたはテーブルに対する許可を付与します。付与はこのような形式になります。

  Grant (アクセス許可) to (プリンシパル) on (リソース) [with grant option]

  grant オプションは、付与対象者が他のプリンシパルに許可を付与することを可能にします。
- タグベースのアクセスコントロール – この方式では、Data Catalog のデータベース、テーブル、および列に 1 つまたは複数の LF タグを割り当てて、1 つまたは複数の LF タグに対するアクセス許可をプリンシパルに付与します。各 LF タグは、`department=sales` のようなキーと値のペアです。Data Catalog リソースの LF タグと一致する LF タグを持つプリンシパルが、そのリソースにアクセスできます。この方式は、多数のデータベースとテーブルを持つデータレイクに推奨されます。これは、「Lake Formation のタグベースのアクセス制御」で詳しく説明されています。

プリンシパルがリソースに対して持っている許可は、両方の方式によって付与された許可を結合したものです。

以下の表は、Data Catalog リソースに対して利用できる Lake Formation 許可の要約です。列の見出しは、許可が付与されるリソースを示しています。

カタログ	データベース	テーブル
CREATE_DATABASE	CREATE_TABLE	ALTER
	ALTER	DROP
	DROP	DESCRIBE
	DESCRIBE	SELECT*
		INSERT*
		DELETE*

例えば、データベースに対する CREATE_TABLE 許可が付与されるとします。これは、プリンシパルがそのデータベース内にテーブルを作成できることを意味します。

アスタリスク (*) が付いた許可は Data Catalog リソースについて付与されますが、基盤となるデータに適用されます。例えば、メタデータテーブルに対する DROP 許可は、Data Catalog からテーブルをドロップできるようにします。一方で、同じテーブルについて付与された DELETE 許可は、Amazon S3 内にあるテーブルの基盤となるデータを、SQL DELETE 文などを使用して削除できるようにします。これらの許可があれば、Lake Formation コンソールでテーブルを表示したり、AWS Glue API を使用してテーブルに関する情報を取得したりすることもできます。したがって、SELECT、INSERT、および DELETE は、Data Catalog 許可とデータアクセス許可の両方になります。

テーブルに対する SELECT を付与するときは、1 つ、または複数の列を包含する、または除外するフィルターを追加できます。これは、メタデータテーブル列に対する細粒度のアクセスコントロールを可能にして、統合されたサービスのユーザーがクエリを実行するときに表示される列を制限します。この機能は、IAM ポリシーのみを使用して利用することはできません。

Super という名前の特別な許可もあります。この Super 許可は、プリンシパルが、許可の対象であるデータベースまたはテーブルで、サポートされているすべての Lake Formation 操作を実行できるようにします。この許可は、他の Lake Formation 許可と共存できます。例えば、メタデータテーブルに対する Super、SELECT、および INSERT を付与することができます。プリンシパルは、サ

ポートされているすべてのアクションをテーブルで実行でき、Super 許可を取り消しても、SELECT と INSERT 許可は残ります。

各許可の詳細については、「Lake Formation 許可のリファレンス」を参照してください。

> **Important**
>
> 別のユーザーが作成した Data Catalog テーブルを表示するには、そのテーブルに対する Lake Formation 許可が、少なくとも 1 つ付与されている必要があります。テーブルに対する許可が少なくとも 1 つ付与されている場合は、テーブルが含まれているデータベースも表示することができます。

Data Catalog 許可は、Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して付与または取り消すことができます。以下は、retailデータベースにテーブルを作成するアクセスdatalake_user1許可をユーザーに付与する AWS CLI コマンドの例です。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "CREATE_TABLE" --resource '{ "Database": {"Name":"retail"}}'
```

以下は、Lake Formation 許可による細粒度のアクセスコントロールを補完する粗粒度のアクセスコントロール IAM ポリシーの例です。これは、任意のメタデータデータベースまたはテーブルに対するすべての操作を許可します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:*Database*",
                "glue:*Table*",
                "glue:*Partition*"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

次の例も粗粒度ですが、制限が多少厳しくなります。これは、指定されたアカウントとリージョン内の Data Catalog にある、すべてのメタデータデータベースおよびテーブルに対する読み取り専用操作を許可します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTables",
                "glue:SearchTables",
                "glue:GetTable",
                "glue:GetDatabase",
                "glue:GetDatabases"
            ],
            "Resource": "arn:aws:glue:us-east-1:111122223333:*"
        }
    ]
}
```

これらのポリシーを、IAM ベースの細粒度のアクセスコントロールを実装する以下のポリシーと比較してください。これは、指定されたアカウントとリージョン内の顧客関係管理 (CRM) メタデータデータベースにあるテーブルのサブセットのみに対する許可を付与します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTables",
                "glue:SearchTables",
```

```
                "glue:GetTable",
                "glue:GetDatabase",
                "glue:GetDatabases"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:catalog",
                "arn:aws:glue:us-east-1:111122223333:database/CRM",
                "arn:aws:glue:us-east-1:111122223333:table/CRM/P*"
            ]
        }
    ]
}
```

粗粒度のアクセスコントロールポリシーの追加例については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

基盤となるデータのアクセスコントロール

統合された AWS サービスが、アクセスが制御されている Amazon S3 ロケーションのデータへのアクセスをリクエストすると AWS Lake Formation、Lake Formation はデータにアクセスするための一時的な認証情報を提供します。

Amazon S3 の場所にある基盤となるデータへのアクセスを Lake Formation が制御できるようにするには、Lake Formation にその場所を登録します。

Amazon S3 ロケーションを登録したら、以下の Lake Formation 許可の付与を開始できます。

- そのロケーションをポイントする Data Catalog テーブルに対するデータアクセス許可 (SELECT、INSERT、および DELETE)。
- そのロケーションに対するデータロケーション許可。

Lake Formation のデータロケーション許可は、特定の Amazon S3 ロケーションをポイントする Data Catalog リソースを作成する機能を制御します。データロケーション許可は、データレイク内のロケーションのセキュリティをさらに強化します。プリンシパルに CREATE_TABLE または ALTER 許可を付与するときは、プリンシパルがメタデータテーブルの作成または変更を実行できるロケーションを制限するためのデータロケーション許可も付与します。

Amazon S3 ロケーションは、バケット、またはバケット下のプレフィックスで、個々の Amazon S3 オブジェクトではありません。

データロケーション許可は、Lake Formation コンソール、API、または AWS CLIを使用してプリンシパルに付与することができます。付与の一般的な形式は以下のとおりです。

```
grant DATA_LOCATION_ACCESS to principal on S3 location [with grant option]
```

`with grant option` を含めると、付与対象者は他のプリンシパルに許可を付与することができます。

Lake Formation のアクセス許可は常に、きめ細かなアクセスコントロールのための AWS Identity and Access Management (IAM) アクセス許可と組み合わせて機能することを覚えておいてください。基盤となる Amazon S3 データに対する読み取り/書き込み許可では、IAM 許可が以下のように付与されます。

ロケーションを登録するときは、そのロケーションに対する読み取り/書き込み許可を付与する IAM ロールを指定します。Lake Formation は、統合 AWS サービスに一時的な認証情報を提供するときに、そのロールを引き受けます。典型的なロールには、以下のようなポリシーがアタッチされている場合があります。このポリシーの登録済みロケーションはバケット `awsexamplebucket` です。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
```

```
            ]
        }
    ]
}
```

Lake Formation は、このようなポリシーを自動的に作成するために登録時に使用できる、サービスリンクロールを提供します。詳細については、「Lake Formation のサービスリンクロールの使用」を参照してください。

このため、Amazon S3 ロケーションの登録によって、そのロケーションに対する必要な IAM `s3:` 許可が付与され、この許可は、ロケーションの登録に使用されたロールによって指定されます。

> **Important**
>
> [Requester pays] (リクエスタ支払い) が有効になっている Amazon S3 バケットの登録は避けてください。Lake Formation に登録されたバケットの場合、バケットの登録に使用されるロールは常にリクエスト元であると見なされます。バケットが別の AWS アカウントからアクセスされた場合、ロールがバケット所有者と同じアカウントに属している場合、バケット所有者はデータアクセスに対して課金されます。

基盤となるデータへの読み取り/書き込みアクセスの場合、プリンシパルには、Lake Formation アクセス許可に加えて以下の `lakeformation:GetDataAccess` IAM アクセス許可も必要になります。この許可があると、Lake Formation がデータにアクセスするための一時的な認証情報のリクエストを承諾します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "lakeformation:GetDataAccess",
            "Resource": "*"
        }
    ]
}
```

上記のポリシーでは、リソースパラメータを「*」(すべて) に設定する必要があります。このアクセス許可に他のリソースを指定することはサポートされていません。この設定により、Lake Formation はデータレイク環境全体のデータアクセスを効率的に管理できます。

> **Note**
>
> Amazon Athena では、ユーザーに `lakeformation:GetDataAccess` アクセス許可が必要です。他の統合サービスでは、基盤となる実行ロールに `lakeformation:GetDataAccess` アクセス許可が必要です。

この許可は、「Lake Formation のペルソナと IAM 許可のリファレンス」で提案されているポリシーに含まれています。

要約すると、Lake Formation プリンシパルが Lake Formation 許可でアクセス制御されている基盤となるデータに対する読み取りと書き込みを実行できるようにするには、以下が必要になります。

- データが含まれる Amazon S3 ロケーションを Lake Formation に登録します。
- 基盤となるデータのロケーションをポイントする Data Catalog テーブルを作成するプリンシパルにデータロケーション許可があること。
- 基盤となるデータに対する読み取りと書き込みを実行するプリンシパルに、基盤となるデータのロケーションをポイントする Data Catalog テーブルに対する Lake Formation データアクセス許可があること。
- 基礎となるデータロケーションが Lake Formation に登録されているとき、基礎となるデータを読み書きするプリンシパルには `lakeformation:GetDataAccess` IAM アクセス許可が必要です。

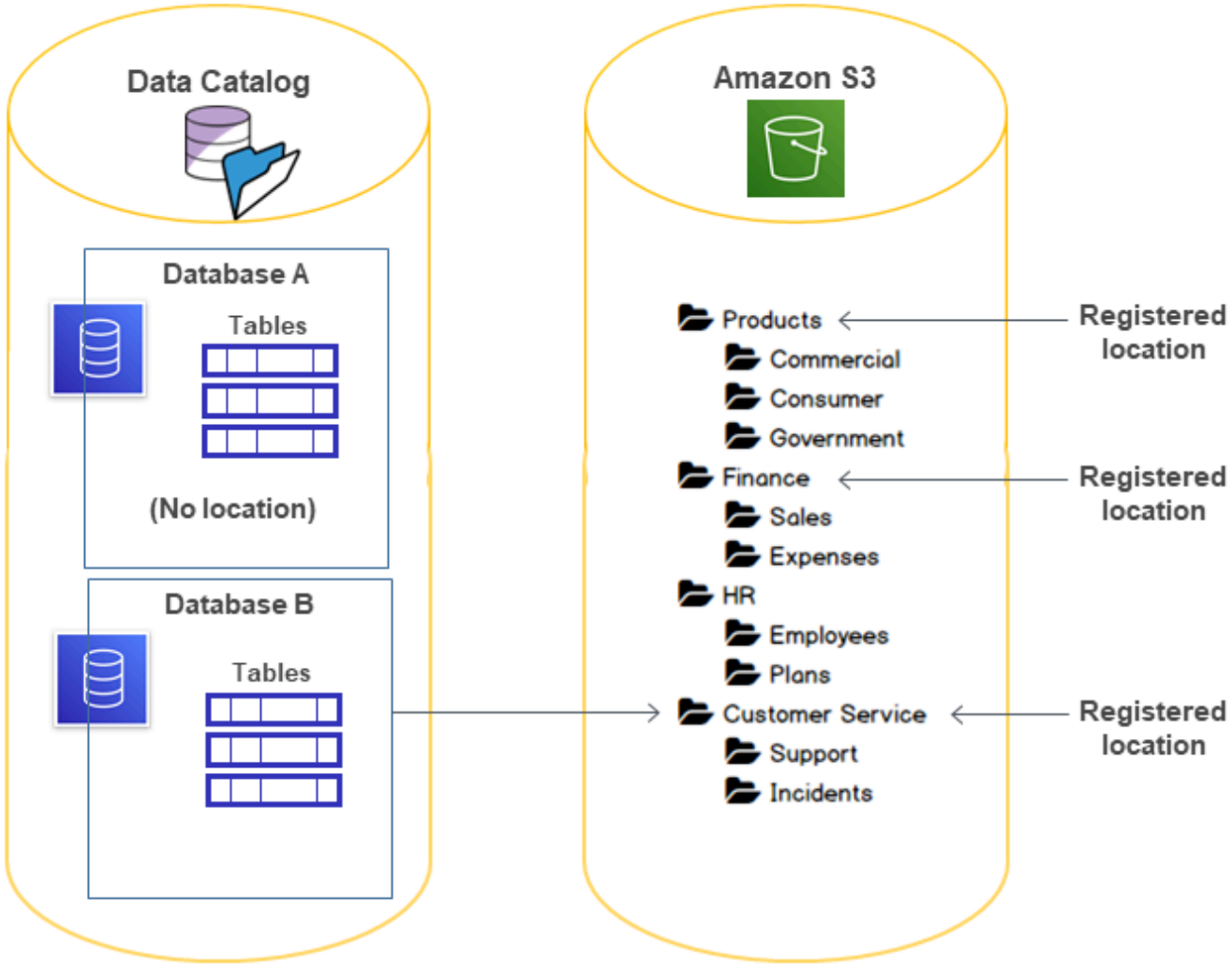
> **Note**
>
> ユーザーが IAM または Amazon S3 ポリシーを通して Amazon S3 ロケーションへのアクセス権を得ている場合、Lake Formation 許可モデルは、Amazon S3 API またはコンソール経由でのそれらのロケーションへのアクセスを阻止しません。IAM ポリシーをプリンシパルにアタッチして、このアクセスをブロックすることができます。

データロケーションアクセス許可の詳細

データロケーション許可は、Data Catalog データベースとテーブルに対する作成および更新操作の結果を制御します。ルールは以下のとおりです。

- プリンシパルが Amazon S3 ロケーションを指定するデータベースまたはテーブルを作成または更新するには、そのロケーションに対する明示的または黙示的なデータロケーション許可を持っている必要があります。
- 明示的なアクセス許可DATA_LOCATION_ACCESSは、コンソール、API、または を使用して付与されます AWS CLI。
- 黙示的な許可は、登録されたロケーションをポイントするロケーションプロパティがデータベースにあり、プリンシパルがそのデータベースに対する CREATE_TABLE 許可を持っていて、プリンシパルがそのロケーションまたは子ロケーションでテーブルを作成しようとするときに付与されます。
- そのロケーションに対するデータロケーション許可がプリンシパルに付与されている場合、プリンシパルはすべての子ロケーションに対するデータロケーション許可を持っています。
- プリンシパルに、基盤となるデータに対する読み取り/書き込み操作を実行するためのデータロケーション許可は必要ありません。SELECT または INSERT データアクセス許可があれば十分です。データロケーション許可は、そのロケーションをポイントする Data Catalog リソースの作成のみに適用されます。
- パーティションオペレーション (CreatePartition、BatchCreatePartition、UpdatePartition) は、パーティションロケーションが Lake Formation に登録済みのテーブルロケーションの子である場合、データロケーションのアクセス許可を必要としません。

以下の図にあるシナリオを考えてみましょう。



この図では、以下のようになっています。

- Amazon S3 バケット Products、Finance、および Customer Service が Lake Formation に登録されている。
- Database A にはロケーションプロパティがなく、Database B には Customer Service バケットをポイントするロケーションプロパティがある。
- ユーザー datalake_user が両方のデータベースに対する CREATE_TABLE を持っている。
- ユーザー datalake_user には、Products バケットのみに対するデータロケーション許可が付与されている。

以下は、ユーザー datalake_user が特定のロケーションで特定のデータベース内にカタログテーブルを作成しようとする場合の結果です。

datalake_user がテーブルを作成しようとするロケーション

データベースとロケーション	成功または失敗	理由
Finance/Sales でのデータベース A	失敗	データロケーション許可がない
Products でのデータベース A	成功	データロケーション許可がある
HR/Plans でのデータベース A	成功	ロケーションが登録されていない
Customer Service/Incidents でのデータベース B	成功	データベースに Customer Service のロケーションプロパティがある

詳細については、次を参照してください。

- データレイクへの Amazon S3 ロケーションの追加
- Lake Formation 許可のリファレンス
- Lake Formation のペルソナと IAM 許可のリファレンス

Lake Formation のペルソナと IAM 許可のリファレンス

このセクションでは、Lake Formation の推奨されるペルソナと、これらのペルソナに推奨される AWS Identity and Access Management (IAM) 許可を一覧表示します。Lake Formation 許可については、「the section called “Lake Formation 許可のリファレンス”」を参照してください。

AWS Lake Formation ペルソナ

次の表に、推奨される AWS Lake Formation ペルソナを示します。

Lake Formation のペルソナ

ペルソナ	説明
IAM 管理者 (スーパーユーザー)	(必須) IAM ユーザーとロールを作成できるユーザーです。`AdministratorAccess` AWS 管理ポリシーがあります。すべての Lake Formation リソースに対するすべての許可を持っています。データレイク管理者を追加できます。データレイク管理者としても指定されている場合を除き、Lake Formation 許可を付与することはできません。
データレイク管理者	(必須) Amazon S3 ロケーションの登録、データカタログへのアクセス、データベースの作成、ワークフローの作成と実行、他のユーザーへの Lake Formation アクセス許可の付与、 AWS CloudTrail ログの表示を行うことができるユーザー。IAM 許可の数は IAM 管理者よりも少ないですが、データレイクを管理するには十分な許可を持っています。他のデータレイク管理者を追加することはできません。
読み取り専用管理者	(オプション) プリンシパル、データカタログリソース、アクセス許可、および AWS CloudTrail ログを表示できますが、更新するアクセス許可を持たないユーザー。
データエンジニア	(オプション) データベースの作成、クローラとワークフローの作成と実行、およびクローラとワークフローが作成する Data Catalog テーブルに対する Lake Formation 許可の付与を実行できるユーザーです。すべてのデータエンジニアをデータベース作成者にすることが推奨されます。詳細については、「データベースを作成する」を参照してください。
データアナリスト	(オプション) Amazon Athenaなどを使用して、データレイクに対するクエリを実行できるユーザーです。クエリを実行するために十分な許可のみを持っています。
ワークフローロール	(必須) ユーザーに代わってワークフローを実行するロールです。このロールは、ブループリントからワークフローを作成するときに指定します。

> **Note**
>
> Lake Formation では、データベースの作成後に追加されたデータレイク管理者はアクセス許可を付与できますが、SELECT や DESCRIBE などのデータアクセス許可が自動的に付与されるわけではありません。データベースを作成する管理者は、それらのデータベースに対する SUPER アクセス許可を受け取ります。この動作は意図したものであり、すべての管理者が自身に必要なアクセス許可を付与できますが、これらのアクセス許可が既存のリソースに自動的に適用されることはありません。したがって、管理者は管理者権限が割り当てられる前に存在していたデータベースへのアクセスを自分自身に明示的に付与する必要があります。

AWS Lake Formation の マネージドポリシー

AWS 管理ポリシーとインラインポリシー AWS Lake Formation を使用して、 の操作に必要な AWS Identity and Access Management (IAM) アクセス許可を付与できます。Lake Formation では、次の AWS 管理ポリシーを使用できます。

AWS マネージドポリシー:AWSLakeFormationDataAdmin

AWSLakeFormationDataAdmin ポリシーは、データレイクを管理する AWS Glue ための AWS Lake Formation や などの関連サービスへの管理アクセスを許可します。

ユーザー、グループおよびロールに `AWSLakeFormationDataAdmin` をアタッチできます。

アクセス許可の詳細

- `CloudTrail` – プリンシパルに AWS CloudTrail ログの表示を許可します。これは、データレイクの設定エラーを確認するために必要です。
- `Glue` — プリンシパルに対して、Data Catalog 内のメタデータテーブルおよびデータベースの表示、作成、更新を許可します。これには、`Get`、`List`、`Create`、`Update`、`Delete`、`Search` で始まる API オペレーションが含まれます。これはデータレイクテーブルのメタデータを管理するために必要です。
- `IAM` — プリンシパルに対して、IAM ユーザー、ロール、およびロールにアタッチされたポリシーに関する情報の取得を許可します。これは、データ管理者が IAM ユーザーおよびロールを確認して表示し、Lake Formation のアクセス許可を付与するために必要です。
- `Lake Formation` — データレイク管理者に対して、データレイクを管理するために必要な Lake Formation のアクセス許可を付与します。

- S3 — プリンシパルに対して、Amazon S3 バケットとその場所に関する情報を取得し、データレイクのデータロケーションを設定することを許可します。

```
"Statement": [
        {
            "Sid": "AWSLakeFormationDataAdminAllow",
            "Effect": "Allow",
            "Action": [
                "lakeformation:*",
                "cloudtrail:DescribeTrails",
                "cloudtrail:LookupEvents",
                "glue:CreateCatalog",
  "glue:UpdateCatalog",
                "glue:DeleteCatalog",
  "glue:GetCatalog",
         "glue:GetCatalogs",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetConnections",
                "glue:SearchTables",
                "glue:GetTable",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTableVersions",
                "glue:GetPartitions",
                "glue:GetTables",
                "glue:ListWorkflows",
                "glue:BatchGetWorkflows",
                "glue:DeleteWorkflow",
                "glue:GetWorkflowRuns",
                "glue:StartWorkflowRun",
                "glue:GetWorkflow",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets",
                "s3:GetBucketAcl",
                "iam:ListUsers",
                "iam:ListRoles",
```

```
                "iam:GetRole",
                "iam:GetRolePolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSLakeFormationDataAdminDeny",
            "Effect": "Deny",
            "Action": [
                "lakeformation:PutDataLakeSettings"
            ],
                "Resource": "*"
        }
    ]
}
```

> **Note**
>
> AWSLakeFormationDataAdmin ポリシーは、データレイク管理者に必要なすべての許可を付与しません。ワークフローの作成と実行、およびサービスリンクロール AWSServiceRoleForLakeFormationDataAccess を使用したロケーションの登録には、追加の許可が必要です。詳細については、「データレイク管理者を作成する」および「Lake Formation のサービスリンクロールの使用」を参照してください。

AWS マネージドポリシー:AWSLakeFormationCrossAccountManager

AWSLakeFormationCrossAccountManager ポリシーは、Lake Formation を介して AWS Glue リソースへのクロスアカウントアクセスを提供し、 AWS Organizations や などの他の必要なサービスへの読み取りアクセスを許可します AWS RAM。

ユーザー、グループおよびロールに AWSLakeFormationCrossAccountManager をアタッチできます。

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。

- Glue — プリンシパルに対して、アクセス制御用の Data Catalog リソースポリシーの設定または削除を許可します。

- Organizations — プリンシパルに対して、組織のアカウントおよび組織単位 (OU) 情報の取得を許可します。
- ram:CreateResourceShare — プリンシパルに対して、リソース共有の作成を許可します。
- ram:UpdateResourceShare —プリンシパルに対して、指定したリソース共有の一部のプロパティの変更を許可します。
- ram:DeleteResourceShare — プリンシパルに対して、指定したリソース共有の削除を許可します。
- ram:AssociateResourceShare — プリンシパルに対して、指定したプリンシパルのリストとリソースのリストをリソース共有に追加することを許可します。
- ram:DisassociateResourceShare — プリンシパルに対して、指定したプリンシパルまたはリソースを、指定したリソース共有への参加から除外することを許可します。
- ram:GetResourceShares — プリンシパルに対して、ユーザー自身が所有しているか、ユーザー自身と共有しているリソース共有に関する詳細を取得することを許可します。
- ram:RequestedResourceType — プリンシパルに対して、リソースタイプ (データベース、テーブル、またはカタログ) の取得を許可します。
- AssociateResourceSharePermission – プリンシパルがリソース共有に含まれるリソースタイプの AWS RAM アクセス許可を追加または置換できるようにします。リソース共有内のリソースタイプごとに、1 つのアクセス許可のみを関連付けることができます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "AllowCreateResourceShare",
            "Effect": "Allow",
            "Action": [
                "ram:CreateResourceShare"
            ],
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "ram:RequestedResourceType": [
                        "glue:Table",
                        "glue:Database",
                        "glue:Catalog"
                    ]
```

```
                }
            }
        },
        {
            "Sid": "AllowManageResourceShare",
            "Effect": "Allow",
            "Action": [
                "ram:UpdateResourceShare",
                "ram:DeleteResourceShare",
                "ram:AssociateResourceShare",
                "ram:DisassociateResourceShare",
                "ram:GetResourceShares"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ram:ResourceShareName": [
                        "LakeFormation*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowManageResourceSharePermissions",
            "Effect": "Allow",
            "Action": [
                "ram:AssociateResourceSharePermission"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ram:PermissionArn": [
                        "arn:aws:ram::aws:permission/AWSRAMLFEnabled*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowXAcctManagerPermissions",
            "Effect": "Allow",
            "Action": [
                "glue:PutResourcePolicy",
                "glue:DeleteResourcePolicy",
                "organizations:DescribeOrganization",
```

```
                "organizations:DescribeAccount",
                "ram:Get*",
                "ram:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowOrganizationsPermissions",
            "Effect": "Allow",
            "Action": [
                "organizations:ListRoots",
                "organizations:ListAccountsForParent",
                "organizations:ListOrganizationalUnitsForParent"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS マネージドポリシー:AWSGlueConsoleFullAccess

AWSGlueConsoleFullAccess ポリシーは、ポリシーがアタッチされている ID が を使用する場合、AWS Glue リソースへのフルアクセスを許可します AWS マネジメントコンソール。このポリシーで指定されたリソースの命名規則に従った場合、ユーザーは完全なコンソール機能を使用できます。このポリシーは通常、 AWS Glue コンソールのユーザーにアタッチされます。

さらに、AWS Glue と Lake Formation は、Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Simple Storage Service (Amazon S3)、および Amazon CloudWatch などの関連サービスへのアクセスを許可するために、サービスロール `AWSGlueServiceRole` を引き受けます。

AWS managed policy:LakeFormationDataAccessServiceRolePolicy

このポリシーは、ユーザーのリクエストに応じてサービスがリソースに対してアクションを実行することを許可する、`ServiceRoleForLakeFormationDataAccess` というサービスリンクロールにアタッチされます。このポリシーを IAM ID にアタッチすることはできません。

このポリシーにより、 Amazon Athena や Amazon Redshift などの Lake Formation 統合 AWS サービスが、サービスにリンクされたロールを使用して Amazon S3 リソースを検出できるようになります。

詳細については、Lake Formation のサービスリンクロールの使用 を参照してください。

アクセス許可の詳細

このポリシーには、次の許可が含まれています。

- s3:ListAllMyBuckets - 認証されたリクエスト送信者が所有するすべてのバケットのリストを返します。

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "LakeFormationDataAccessServiceRolePolicy",
   "Effect": "Allow",
   "Action": [
    "s3:ListAllMyBuckets"
   ],
   "Resource": [
    "arn:aws:s3:::*"
   ]
  }
 ]
}
```

AWS 管理ポリシーに対する Lake Formation の更新

このサービスがこれらの変更の追跡を開始してからの Lake Formation の AWS マネージドポリシーの更新に関する詳細を表示します。

変更	説明	日付
Lake Formation が AWSLakeFormationCrossAccountManager	Lake Formation は、StringLike 条件演算子を、IAM による ARN 形式チェックの実行を可能にする ArnLike 演算子に置き換えて、AWSLakeFo	2025 年 1 月

変更	説明	日付
ポリシーを更新しました。	rmationCrossAccountManager ポリシーを強化しました。	
Lake Formation が AWSLakeFormationDataAdmin ポリシーを更新しました。	Lake Formation は、マルチカタログ機能の一部として次の AWS Glue Data Catalog CRUD API を追加して、AWSLakeFormationDataAdmin ポリシーを強化しました。 • glue:CreateCatalog • glue:UpdateCatalog • glue:DeleteCatalog • glue:GetCatalog • glue:GetCatalog この管理ポリシーの変更は、Lake Formation 管理者ペルソナにこれらの新しいオペレーションに対する IAM アクセス許可がデフォルトで付与されるようにするためです。	2024 年 12 月
Lake Formation が AWSLakeFormationCrossAccountManager ポリシーを更新しました。	Lake Formation で AWSLakeFormationCrossAccountManager ポリシーが強化され、ポリシーステートメントに Sid 要素が追加されました。	2024 年 3 月
Lake Formation が AWSLakeFormationDataAdmin ポリシーを更新しました。	Lake Formation で AWSLakeFormationDataAdmin ポリシーが強化され、ポリシーステートメントに Sid 要素が追加され、余分なアクションが削除されました。	2024 年 3 月

変更	説明	日付
Lake Formation が `LakeFormationDataAccessServiceRolePolicy` ポリシーを更新しました。	Lake Formation で LakeFormationDataAccessServiceRolePolicy ポリシーが強化され、ポリシーステートメントに Sid 要素が追加されました。	2024 年 2 月
Lake Formation が `AWSLakeFormationCrossAccountManager` ポリシーを更新しました。	Lake Formation では、ハイブリッドアクセスモードでのクロスアカウントデータ共有を可能にする新しいアクセス許可を追加し、AWSLakeFormationCrossAccountManager ポリシーを強化しました。	2023 年 10 月
Lake Formation が `AWSLakeFormationCrossAccountManager` ポリシーを更新しました。	Lake Formation は AWSLakeFormationCrossAccountManager ポリシーを強化し、リソースの最初の共有時に、受信者アカウントごとに 1 つのリソース共有のみを作成するようになりました。以降に同じアカウントで共有されるすべてのリソースは、同じリソース共有にアタッチされます。	2022 年 5 月 6 日
Lake Formation が変更の追跡を開始しました。	Lake Formation は AWS 、管理ポリシーの変更の追跡を開始しました。	2022 年 5 月 6 日

ペルソナに推奨される許可

以下は、各ペルソナに推奨される許可です。IAM 管理者であるユーザーは、すべてのリソースに対するすべての許可を持っているため、ここのは含まれていません。

トピック

- データレイク管理者の許可
- 読み取り専用管理者のアクセス許可
- データエンジニアの許可

- データアナリストの許可
- ワークフローロールの許可

データレイク管理者の許可

> **Important**
>
> 次のポリシーでは、*<account-id>* を有効な AWS アカウント番号に置き換え、*<workflow_role>* を、「」で定義されているように、ワークフローを実行するアクセス許可を持つロールの名前に置き換えますワークフローロールの許可。

ポリシータイプ	ポリシー
AWS マネージドポリシー	• AWSLakeFormationDataAdmin • LakeFormationDataAccessServiceRolePolicy (サービスリンクロールポリシー) • AWSGlueConsoleFullAccess (オプション) • CloudWatchLogsReadOnlyAccess (オプション) • AWSLakeFormationCrossAccountManager (オプション) • AmazonAthenaFullAccess (オプション) オプションの AWS 管理ポリシーの詳細については、「」を参照してくださいthe section called “データレイク管理者を作成する”。

インラインポリシー (Lake Formation サービスリンクロールの作成用)	<pre>{
	    "Version": "2012-10-17",
	    "Statement": [
	        {
	            "Effect": "Allow",
	            "Action": "iam:CreateServiceLinkedRole",
	            "Resource": "*",
	            "Condition": {</pre>

ポリシータイプ	ポリシー

	"StringEquals": {
	"iam:AWSServiceName": "lakeformation.amazonaws.com"
	}
	}
	},
	{
	"Effect": "Allow",
	"Action": [
	"iam:PutRolePolicy"
	],
	"Resource": "arn:aws:iam:: *<account-id>* :role/aws-service-role/lakeformation.amazonaws.com/AWSServiceRoleForLakeFormationDataAccess"
	}
	]
	}

(オプション) インラインポリシー (ワークフローロールのための PassRole ポリシー)。これは、データレイク管理者がワークフローを作成して実行する場合にのみ必要になります。	**{
	"Version":"2012-10-17",
	"Statement": [
	{
	"Sid": "PassRolePermissions",
	"Effect": "Allow",
	"Action": [
	"iam:PassRole"
	],
	"Resource": [
	"arn:aws:iam:: *111122223333* :role/*<workflow_role>* "
	]
	}
	]
	}**

ポリシータイプ	ポリシー

(オプション) インラインポリシー (アカウントがクロスアカウント Lake Formation 許可を付与または受けている場合)。このポリシーは、 AWS RAM リソース共有の招待を承諾または拒否し、組織へのクロスアカウントアクセス許可の付与を有効にするためのものです。 `ram:EnableSharingWithAwsOrganization` は、管理アカウントのデータレイク管理者 AWS Organizations にのみ必要です。	<pre>{
	    "Version":"2012-10-17",
	    "Statement": [
	        {
	            "Effect": "Allow",
	            "Action": [
	                "ram:AcceptResourceShareInvitation",
	                "ram:RejectResourceShareInvitation",
	                "ec2:DescribeAvailabilityZones",
	                "ram:EnableSharingWithAwsOrganization"
	            ],
	            "Resource": "*"
	        }
	    ]
	}</pre>

読み取り専用管理者のアクセス許可

ポリシータイプ	ポリシー

インラインポリシー (ベーシック)	<pre>{
	   "Version":"2012-10-17",
	   "Statement":[
	      {
	         "Effect":"Allow",
	         "Action":[
	            "lakeformation:GetEffectivePermissionsForPath",
	            "lakeformation:ListPermissions",
	            "lakeformation:ListDataCellsFilter",
	            "lakeformation:GetDataCellsFilter",
	            "lakeformation:SearchDatabasesByLFTags",</pre>

ポリシータイプ	ポリシー

```
            "lakeformation:SearchTablesByLFTags"
,
            "lakeformation:GetLFTag",
            "lakeformation:ListLFTags",
            "lakeformation:GetResourceLFTags",
            "lakeformation:ListLakeFormationOpti
ns",
            "cloudtrail:DescribeTrails",
            "cloudtrail:LookupEvents",
            "glue:GetDatabase",
            "glue:GetDatabases",
            "glue:GetConnections",
            "glue:SearchTables",
            "glue:GetTable",
            "glue:GetTableVersions",
            "glue:GetPartitions",
            "glue:GetTables",
            "glue:GetWorkflow",
            "glue:ListWorkflows",
            "glue:BatchGetWorkflows",
            "glue:GetWorkflowRuns",
            "glue:GetWorkflow",
            "s3:ListBucket",
            "s3:GetBucketLocation",
            "s3:ListAllMyBuckets",
            "s3:GetBucketAcl",
            "iam:ListUsers",
            "iam:ListRoles",
            "iam:GetRole",
            "iam:GetRolePolicy"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Deny",
         "Action":[
            "lakeformation:PutDataLakeSettings"
         ],
         "Resource":"*"
      }
   ]
```

ポリシータイプ	ポリシー
	}

データエンジニアの許可

> **Important**
>
> 次のポリシーでは、*<account-id>* を有効な AWS アカウント番号に置き換え、*<workflow_role>* をワークフローロールの名前に置き換えます。

ポリシータイプ	ポリシー
AWS マネージドポリシー	AWSGlueConsoleFullAccess
インラインポリシー (ベーシック)	{ "Version":"2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["lakeformation:GetDataAccess", "lakeformation:GrantPermissions", "lakeformation:RevokePermissions", "lakeformation:BatchGrantPermissions", "lakeformation:BatchRevokePermissions", "lakeformation:ListPermissions", "lakeformation:AddLFTagsToResource", "lakeformation:RemoveLFTagsFromResource", "lakeformation:GetResourceLFTags", "lakeformation:ListLFTags", "lakeformation:GetLFTag", "lakeformation:SearchTablesByLFTags",

ポリシータイプ	ポリシー
	"lakeformation:SearchDatabasesByLFTa gs", "lakeformation:GetWorkUnits", "lakeformation:GetWorkUnitResults", "lakeformation:StartQueryPlanning", "lakeformation:GetQueryState", "lakeformation:GetQueryStatistics"], "Resource": "*" }] }

インラインポリシー (トランザクション内での操作を含む、管理対象テーブルでの操作用)	{
	"Version":"2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"lakeformation:StartTransaction",
	"lakeformation:CommitTransaction",
	"lakeformation:CancelTransaction",
	"lakeformation:ExtendTransaction",
	"lakeformation:DescribeTransaction",
	"lakeformation:ListTransactions",
	"lakeformation:GetTableObjects",
	"lakeformation:UpdateTableObjects",
	"lakeformation:DeleteObjectsOnCancel
	"
	],
	"Resource": "*"
	}
	]
	}

ポリシータイプ	ポリシー

インラインポリシー (Lake Formation のタグベースのアクセス制御 (LF-TBAC) 方式を使用したメタデータアクセス制御用)	{
	"Version":"2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"lakeformation:AddLFTagsToResource",
	"lakeformation:RemoveLFTagsFromResou
	rce",
	"lakeformation:GetResourceLFTags",
	"lakeformation:ListLFTags",
	"lakeformation:GetLFTag",
	"lakeformation:SearchTablesByLFTags"
	,
	"lakeformation:SearchDatabasesByLFTa
	gs"
	],
	"Resource": "*"
	}
	]
	}

インラインポリシー (ワークフローロールのための PassRole ポリシー)	{
	"Version":"2012-10-17",
	"Statement": [
	{
	"Sid": "PassRolePermissions",
	"Effect": "Allow",
	"Action": [
	"iam:PassRole"
	],
	"Resource": [
	"arn:aws:iam:: *111122223
	333* :role/*<workflow_role>* "
	]
	}
	]
	}

データアナリストの許可

ポリシータイプ	ポリシー
AWS マネージドポリシー	AmazonAthenaFullAccess

インラインポリシー (ベーシック)	<pre>{
	    "Version": "2012-10-17",
	    "Statement": [
	        {
	            "Effect": "Allow",
	            "Action": [
	                "lakeformation:GetDataAccess",
	                "glue:GetTable",
	                "glue:GetTables",
	                "glue:SearchTables",
	                "glue:GetDatabase",
	                "glue:GetDatabases",
	                "glue:GetPartitions",
	                "lakeformation:GetResourceLFTags",
	                "lakeformation:ListLFTags",
	                "lakeformation:GetLFTag",
	                "lakeformation:SearchTablesByLFTags",
	                "lakeformation:SearchDatabasesByLFTags"
	
	            ],
	            "Resource": "*"
	        }
	    ]
	}</pre>

(オプション) インラインポリシー (トランザクション内での操作を含む、管理対象テーブルでの操作用)	<pre>{
	    "Version":"2012-10-17",
	    "Statement": [
	        {
	            "Effect": "Allow",
	            "Action": [
	                "lakeformation:StartTransaction",
	                "lakeformation:CommitTransaction",
	                "lakeformation:CancelTransaction",
	                "lakeformation:ExtendTransaction",</pre>

ポリシータイプ	ポリシー

	<pre>"lakeformation:DescribeTransaction",
	"lakeformation:ListTransactions",
	"lakeformation:GetTableObjects",
	"lakeformation:UpdateTableObjects",
	"lakeformation:DeleteObjectsOnCancel
	"
	],
	"Resource": "*"
	}
	]
	}</pre>

ワークフローロールの許可

このロールには、ワークフローを実行するために必要な許可があります。これらの許可を持つロールは、ワークフローを作成するときに指定します。

> **Important**
>
> 次のポリシーでは、*<region>* を有効な AWS リージョン識別子 (例: us-east-1)、*<account-id>* を有効な AWS アカウント番号、*<workflow_role>* をワークフローロールの名前、*<your-s3-cloudtrail-bucket>* を AWS CloudTrail ログへの Amazon S3 パスに置き換えます。

ポリシータイプ	ポリシー
AWS マネージドポリシー	AWSGlueServiceRole

インラインポリシー (データアクセス)	<pre>{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Sid": "Lakeformation",
	"Effect": "Allow",
	"Action": [
	"lakeformation:GetDataAccess",</pre>

ポリシータイプ	ポリシー

	<pre>"lakeformation:GrantPermissions"
	],
	"Resource": "*"
	}
	]
	}</pre>

インラインポリシー (ワークフローロールのための PassRole ポリシー)	<pre>{
	"Version":"2012-10-17",
	"Statement": [
	{
	"Sid": "PassRolePermissions",
	"Effect": "Allow",
	"Action": [
	"iam:PassRole"
	],
	"Resource": [
	"arn:aws:iam:: 111122223333 :role/<workflow_role> "
	]
	}
	]
	}</pre>

インラインポリシー (AWS CloudTrail ログなど、データレイクの外部にデータを取り込む場合)	<pre>{
	"Version":"2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": ["s3:GetObject", "s3:ListBucket"],
	"Resource": ["arn:aws:s3::: <your-s3-cloudtrail-bucket> /*"]
	}
	]
	}</pre>

データレイクのデフォルト設定の変更

との下位互換性を維持するためにAWS Glue、 AWS Lake Formation には以下の初期セキュリティ設定があります。

- 既存の AWS Glue Data Catalog リソースのすべてに対する `Super` 許可がグループ `IAMAllowedPrincipals` に付与されます。
- 新しい Data Catalog リソースには「Use only IAM access control」(IAM アクセス制御のみを使用する) 設定が有効になっています。

これらの設定により、データカタログリソースと Amazon S3 ロケーションへのアクセスは、 AWS Identity and Access Management (IAM) ポリシーによってのみ制御されます。個々の Lake Formation 許可は適用されません。

`IAMAllowedPrincipals` グループには、IAM ポリシーによって Data Catalog リソースへのアクセスを許可される IAM ユーザーとロールが含まれます。この `Super` 許可は、プリンシパルが、許可の対象であるデータベースまたはテーブルで、サポートされているすべての Lake Formation 操作を実行できるようにします。

Data Catalog リソース (データベースおよびテーブル) へのアクセスが Lake Formation 許可によって管理されるようにセキュリティ設定を変更するには、以下を実行します。

1. 新しいリソースに対するデフォルトのセキュリティ設定を変更する。手順については、「デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する」を参照してください。
2. 既存の Data Catalog リソースに対する設定を変更する。手順については、「AWS Lake Formation モデルへのAWS Glueデータアクセス許可のアップグレード」を参照してください。

Lake Formation の **`PutDataLakeSettings`** API 操作を使用したデフォルトセキュリティ設定の変更

デフォルトのセキュリティ設定は、Lake Formation の PutDataLakeSettings API オペレーションを使用して変更することもできます。このアクションは、オプションのカタログ ID と DataLakeSettings 構造を引数として使用します。

Lake Formation によるメタデータと基盤となるデータのアクセス制御を新しいデータベースとテーブルに適用するには、`DataLakeSettings` 構造を以下のようにコード化します。

> **Note**
>
> *<AccountID>* を有効な AWS アカウント ID に、*<Username>* を有効な IAM ユーザー名に置き換えます。複数のユーザーをデータレイク管理者として指定できます。

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [
            {
                "DataLakePrincipalIdentifier":
 "arn:aws:iam::<AccountId>:user/<Username>"
            }
        ],
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": []
    }
}
```

この構造は、以下のようにコード化することもできます。`CreateDatabaseDefaultPermissions` または `CreateTableDefaultPermissions` パラメータを省略することは、空のリストを渡すことに相当します。

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [
            {
                "DataLakePrincipalIdentifier":
 "arn:aws:iam::<AccountId>:user/<Username>"
            }
        ]
    }
}
```

このアクションは実質的に、`IAMAllowedPrincipals` グループから新しいデータベースとテーブルに対するすべての Lake Formation 許可を取り消します。この設定は、データベースを作成するときに上書きすることができます。

IAM のみによるメタデータと基盤となるデータのアクセス制御を新しいデータベースとテーブルに適用するには、`DataLakeSettings` 構造を以下のようにコード化します。

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [
            {
                "DataLakePrincipalIdentifier":
 "arn:aws:iam::<AccountId>:user/<Username>"
            }
        ],
        "CreateDatabaseDefaultPermissions": [
            {
                "Principal": {
                    "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                },
                "Permissions": [
                    "ALL"
                ]
            }
        ],
        "CreateTableDefaultPermissions": [
            {
                "Principal": {
                    "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                },
                "Permissions": [
                    "ALL"
                ]
            }
        ]
    }
}
```

これは、新しいデータベースとテーブルに対する Super Lake Formation 許可を IAMAllowedPrincipals グループに付与します。この設定は、データベースを作成するときに上書きすることができます。

> **Note**
>
> 前述の DataLakeSettings 構造では、DataLakePrincipalIdentifier に許可される値は IAM_ALLOWED_PRINCIPALS のみで、Permissions に許可される値は ALL のみです。

黙示的な Lake Formation 許可

AWS Lake Formation は、データレイク管理者、データベース作成者、およびテーブル作成者に次の暗黙的なアクセス許可を付与します。

データレイク管理者

- 別のアカウントから別のプリンシパルに直接共有されているリソースを除き、データカタログ内のすべての Describe リソースにアクセスできます。管理者からこのアクセス権を取り消すことはできません。
- データレイク全体に対するデータロケーション許可があります。
- Data Catalog 内の任意のリソースへのアクセス権を任意のプリンシパル (自分自身を含む) に付与する、またはそれらをまたは取り消すことができます。管理者からこのアクセス権を取り消すことはできません。
- Data Catalog にデータベースを作成できます。
- データベースを作成する許可を別のユーザーに付与できます。

> **Note**
>
> データレイク管理者が Amazon S3 ロケーションを登録できるのは、それを実行するための IAM 許可を持っている場合に限定されます。本ガイドで推奨されているデータレイク管理者ポリシーは、これらの許可を付与します。また、データレイク管理者には、データベースをドロップする、または他のユーザーが作成したテーブルを変更/ドロップするための黙示的な許可はありませんが、それらを実行する許可を自分自身に付与することが可能です。

データレイク管理者の詳細については、「データレイク管理者を作成する」を参照してください。

カタログ作成者

- 作成するカタログに対するすべてのカタログアクセス許可を持ち、カタログ内に作成するデータベースとテーブルに対するアクセス許可を持ち、カタログ内にデータベースとテーブルを作成するアクセス許可を同じ AWS アカウントの他のプリンシパルに付与できます。AWSLakeFormationCrossAccountManager AWS 管理ポリシーも持つカタログ作成者は、カタログに対するアクセス許可を他の AWS アカウントまたは組織に付与できます。

データレイク管理者は、Lake Formation コンソールまたは API を使用してカタログ作成者を指定することができます。

> **Note**
>
> カタログ作成者には、他のユーザーによってカタログ内に作成されるデータベースとテーブルに対する黙示的なアクセス許可はありません。

カタログの作成の詳細については、「へのデータの取り込み AWS Glue Data Catalog」を参照してください。

データベース作成者

- 作成するデータベースに対するすべてのデータベースアクセス許可を持ち、データベースに作成するテーブルに対するアクセス許可を持ち、同じ AWS アカウントの他のプリンシパルにデータベースにテーブルを作成するアクセス許可を付与できます。`AWSLakeFormationCrossAccountManager` AWS マネージドポリシーも持つデータベース作成者は、データベースに対するアクセス許可を他の AWS アカウントまたは組織に付与できます。

  データレイク管理者は、Lake Formation コンソールまたは API を使用してデータベース作成者を指定することができます。

  > **Note**
  >
  > データベース作成者に、他のユーザーがデータベース内に作成するテーブルに対する黙示的な許可はありません。

  詳細については、「データベースを作成する」を参照してください。

テーブル作成者

- 作成するテーブルに対するすべての許可があります。
- 作成するすべてのテーブルに対する許可を同じ AWS アカウント内のプリンシパルに付与できます。
- `AWSLakeFormationCrossAccountManager` AWS 管理ポリシーがある場合は、作成したすべてのテーブルに対するアクセス許可を他の AWS アカウントまたは組織に付与できます。
- 作成するテーブルが含まれるデータベースを表示できます。

Lake Formation 許可のリファレンス

AWS Lake Formation オペレーションを実行するには、プリンシパルに Lake Formation アクセス許可と AWS Identity and Access Management (IAM) アクセス許可の両方が必要です。IAM 許可は通常、「the section called “Lake Formation 許可の概要”」で説明したように、粗粒度のアクセス制御ポリシーを使用して付与します。Lake Formation のアクセス許可は、 コンソール、 API、または AWS Command Line Interface () を使用して付与できますAWS CLI。

Lake Formation 許可を付与または取り消す方法を学ぶには、「the section called “データアクセス許可の付与”」および「the section called “データロケーション許可の付与”」を参照してください。

> **Note**
>
> このセクションの例は、同じ AWS アカウント内のプリンシパルに許可を付与するを説明するものです。クロスアカウント付与の例については、「the section called “クロスアカウントデータ共有”」を参照してください。

リソースタイプ別の Lake Formation 許可

各リソースで利用できる有効な Lake Formation 許可は次のとおりです。

[リソース]	アクセス許可
Catalog	ALL (Super)、スーパーユーザー
	ALTER
	CREATE_DATABASE
	DESCRIBE
	DROP
Database	ALL (Super)
	ALTER
	CREATE_TABLE

[リソース]	アクセス許可	
	DESCRIBE	
	DROP	
Table	ALL (Super)	
	ALTER	
	DELETE	
	DESCRIBE	
	DROP	
	INSERT	
	SELECT	
View	ALL (Super)	
	SELECT	
	DESCRIBE	
	DROP	
Data Catalog	CREATE_DATABASE	
Amazon S3 location	DATA_LOCATION_ACCESS	
LF-Tags	DROP	
	ALTER	
LF-Tag values	ASSOCIATE	
	DESCRIBE	
	GrantWithLFTagExpression	

[リソース]	アクセス許可
LF-Tag policy - Database	ALL (Super)
	ALTER
	CREATE_TABLE
	DESCRIBE
	DROP
LF-Tag policy - Table	ALL (Super)
	ALTER
	DESCRIBE
	DELETE
	DROP
	INSERT
	SELECT
Resource link - Database or Table	DESCRIBE
	DROP
Table with data filters	DESCRIBE
	DROP
	SELECT
Table with column filter	SELECT

トピック

- Lake Formation の許可と取り消し AWS CLI コマンド

- Lake Formation 許可

Lake Formation の許可と取り消し AWS CLI コマンド

このセクションの各アクセス許可の説明には、 AWS CLI コマンドを使用してアクセス許可を付与する例が含まれています。Lake Formation コマンドgrant-permissionsと revoke-permissions AWS CLI コマンドの概要を次に示します。

```
grant-permissions
[--catalog-id <value>]
--principal <value>
--resource <value>
--permissions <value>
[--permissions-with-grant-option <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

```
revoke-permissions
[--catalog-id <value>]
--principal <value>
--resource <value>
--permissions <value>
[--permissions-with-grant-option <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

これらのコマンドの詳しい説明については、「AWS CLI コマンドリファレンス」の「grant-permissions」および「revoke-permissions」を参照してください。このセクションは、`--principal` オプションに関する追加の情報を提供します。

`--principal` オプションの値は、以下のいずれかになります。

- (IAM) ユーザーまたはロールの Amazon リソースネーム AWS Identity and Access Management (ARN)
- Microsoft アクティブディレクトリフェデレーションサービス (AD FS) などの SAML プロバイダー経由で認証するユーザーまたはグループの ARN
- Amazon Quick ユーザーまたはグループの ARN
- クロスアカウントアクセス許可、 AWS アカウント ID、組織 ID、または組織単位 ID の場合

- IAM アイデンティティセンターユーザーまたはグループの場合、IAM アイデンティティセンターユーザーユーザーまたはグループの ARN。

以下は、すべての --principal タイプの構文と例です。

プリンシパルが IAM ユーザー

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:user/<user-name>
```

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1
```

プリンシパルが IAM ロール

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:role/<role-name>
```

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:role/workflowrole
```

プリンシパルが SAML プロバイダー経由で認証するユーザー

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:saml-
provider/<SAMLproviderName>:user/<user-name>
```

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:saml-provider/
idp1:user/datalake_user1
```

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:saml-provider/
AthenaLakeFormationOkta:user/athena-user@example.com
```

プリンシパルが SAML プロバイダー経由で認証するグループ

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:saml-
provider/<SAMLproviderName>:group/<group-name>
```

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:saml-provider/
idp1:group/data-scientists
```

```
--principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:saml-provider/
AthenaLakeFormationOkta:group/my-group
```

プリンシパルは Amazon Quick Enterprise Edition ユーザーです

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:quicksight:<region>:<account-
id>:user/<namespace>/<user-name>
```

> **Note**
>
> *<namespace>* には `default` を指定する必要があります。

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:quicksight:us-
east-1:111122223333:user/default/bi_user1
```

プリンシパルは Amazon Quick Enterprise Edition グループです

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:quicksight:<region>:<account-
id>:group/<namespace>/<group-name>
```

> **Note**
>
> *<namespace>* には default を指定する必要があります。

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:quicksight:us-
east-1:111122223333:group/default/data_scientists
```

プリンシパルは AWS アカウントです

構文:

```
--principal DataLakePrincipalIdentifier=<account-id>
```

例:

```
--principal DataLakePrincipalIdentifier=111122223333
```

プリンシパルが組織

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:organizations::<account-
id>:organization/<organization-id>
```

例:

```
--principal
 DataLakePrincipalIdentifier=arn:aws:organizations::111122223333:organization/o-
abcdefghijkl
```

プリンシパルが組織単位

構文:

```
--principal DataLakePrincipalIdentifier=arn:aws:organizations::<account-
id>:ou/<organization-id>/<organizational-unit-id>
```

例:

```
--principal DataLakePrincipalIdentifier=arn:aws:organizations::111122223333:ou/o-
abcdefghijkl/ou-ab00-cdefghij
```

プリンシパルが IAM アイデンティティセンターの ID ユーザーまたはグループ

例: ユーザー

```
--principal DataLakePrincipalIdentifier=arn:aws:identitystore:::user/<UserID>
```

例: グループ

```
--principal DataLakePrincipalIdentifier=arn:aws:identitystore:::group/<GroupID>
```

プリンシパルが IAM グループ - **IAMAllowedPrincipals**

Lake Formation は、データカタログ内のすべてのデータベースとテーブルに対する Super アクセス許可を、デフォルトで IAMAllowedPrincipals というグループに設定します。このグループアクセス許可がデータベースまたはテーブルに存在する場合、アカウント内のすべてのプリンシパルが、 AWS Glueの IAM プリンシパルポリシーを介してリソースにアクセスできるようになります。これにより、以前に AWS Glueの IAM ポリシーで保護されていたデータカタログリソースを Lake Formation アクセス許可で保護し始めるときに、下位互換性が提供されます。

Lake Formation を使用してデータカタログリソースのアクセス許可を管理する場合、Lake Formation アクセス許可を機能させるには、まずリソースに設定されている IAMAllowedPrincipals アクセス許可を取り消すか、プリンシパルとリソースをハイブリッドアクセスモードにオプトインする必要があります。

例:

```
--principal DataLakePrincipalIdentifier=IAM_Allowed_Principals
```

プリンシパルが IAM グループ - **ALLIAMPrincipals**

データカタログリソースへのアクセス許可を ALLIAMPrincipals グループに付与すると、アカウント内のすべてのプリンシパルが、Lake Formation アクセス許可と IAM アクセス許可を使用してデータカタログリソースにアクセスできるようになります。

例:

```
--principal DataLakePrincipalIdentifier=123456789012:IAMPrincipals
```

Lake Formation 許可

このセクションでは、プリンシパルに付与できる Lake Formation 許可を一覧表示します。

ALTER

許可	付与対象リソース	付与対象に必要な追加の許可
ALTER	DATABASE	glue:UpdateDatabase
ALTER	TABLE	glue:UpdateTable
ALTER	LF-Tag	lakeformation:UpdateLFTag

この許可を持つプリンシパルは、Data Catalog 内のデータベースまたはテーブルのメタデータを変更できます。テーブルの場合は、列スキーマを変更し、列パラメータを追加することができます。メタデータテーブルがポイントする基盤となるデータの列を変更することはできません。

変更されるプロパティが登録済みの Amazon Simple Storage Service (Amazon S3) ロケーションである場合は、プリンシパルが新しいロケーションに対するデータロケーション許可を持っている必要があります。

Example

次の例では、retail AWS アカウント 1111-2222-3333 のデータベースdatalake_user1のユーザーに アクセスALTER許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "ALTER" --resource '{ "Database": {"Name":"retail"}}'
```

Example

以下の例は、データベース retail にあるテーブル inventory に対する ALTER をユーザー datalake_user1 に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
```

```
 --permissions "ALTER" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory"}}'
```

CREATE_DATABASE

許可	付与対象リソース	付与対象に必要な追加の許可
CREATE_DATABASE	Data Catalog	glue:CreateDatabase

この許可を持つプリンシパルは、Data Catalog にメタデータデータベースまたはリソースリンクを作成できます。プリンシパルは、データベースにテーブルを作成することもできます。

Example

次の例ではCREATE_DATABASE、 AWS アカウント 1111-2222-3333 datalake_user1の ユーザーに を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "CREATE_DATABASE" --resource '{ "Catalog": {}}'
```

プリンシパルが Data Catalog にデータベースを作成するときに、基盤となるデータに対する許可は付与されません。以下の追加のメタデータ許可が、これらの許可を他のユーザーに付与する能力と共に付与されます。

- データベース内での CREATE_TABLE
- データベースの ALTER
- データベースの DROP

プリンシパルは、データベースを作成するときにオプションで Amazon S3 ロケーションを指定できます。プリンシパルがデータロケーション許可を持っているかどうかに応じて、CREATE_DATABASE 許可ではデータベースを作成できない場合があります。以下の 3 つのユースケースを念頭に置いておくことが重要です。

データベースの作成ユースケース	必要となる許可
ロケーションプロパティが指定されていない。	CREATE_DATABASE で十分です。

データベースの作成ユースケース	必要となる許可
ロケーションプロパティが指定されており、ロケーションが Lake Formation によって管理されていない (登録されていない)。	CREATE_DATABASE で十分です。
ロケーションプロパティが指定されており、ロケーションが Lake Formation によって管理されている (登録されている)。	CREATE_DATABASE に加えて、指定されたロケーションに対するデータロケーション許可が必要です。

CREATE_TABLE

許可	付与対象リソース	付与対象に必要な追加の許可
CREATE_TABLE	DATABASE	glue:CreateTable

この許可を持つプリンシパルは、指定したデータベース内の Data Catalog にメタデータテーブルまたはリソースリンクを作成できます。

Example

次の例では、 AWS アカウント 1111-2222-3333 のretailデータベースにテーブルを作成するdatalake_user1アクセス許可をユーザーに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "CREATE_TABLE" --resource '{ "Database": {"Name":"retail"}}'
```

プリンシパルが Data Catalog にテーブルを作成すると、そのテーブルに対するすべての Lake Formation 許可が、これらの許可を他のユーザーに付与する能力と共にプリンシパルに付与されます。

クロスアカウント付与

データベース所有者アカウントが受領者アカウントに CREATE_TABLE を付与し、受領者アカウントのユーザーが所有者アカウントのデータベースにテーブルを正常に作成する場合、以下のルールが適用されます。

- 受領者アカウントのユーザーとデータレイク管理者には、このテーブルに対するすべての Lake Formation 許可があり、テーブルに対する許可をアカウント内の他のプリンシパルに付与することができます。所有者アカウントまたはその他のアカウントのプリンシパルに許可を付与することはできません。
- 所有者アカウントのデータレイク管理者は、テーブルに対する許可をアカウント内の他のプリンシパルに付与できます。

データロケーション許可

Amazon S3 ロケーションをポイントするテーブルの作成を試みるときは、データロケーション許可を持っているかどうかに応じて、CREATE_TABLE 許可がテーブルの作成に不十分である場合があります。以下の 3 つのユースケースを念頭に置いておくことが重要です。

テーブルの作成ユースケース	必要となる許可
指定されたロケーションが Lake Formation によって管理されていない (登録されていない)。	CREATE_TABLE で十分です。
指定されたロケーションが Lake Formation によって管理されて (登録されて) おり、それが含まれるデータベースにロケーションプロパティがないか、テーブルロケーションの Amazon S3 プレフィックスではないロケーションプロパティがある。	CREATE_TABLE に加えて、指定されたロケーションに対するデータロケーション許可が必要です。
指定されたロケーションが Lake Formation によって管理されて (登録されて) おり、それが含まれるデータベースに、登録済みで、かつテーブルロケーションの Amazon S3 プレフィックスである ロケーションをポイントするロケーションプロパティがある。	CREATE_TABLE で十分です。

DATA_LOCATION_ACCESS

許可	付与対象リソース	付与対象に必要な追加の許可
DATA_LOCATION_ACCESS	Amazon S3 ロケーション	(このロケーションに対する Amazon S3 許可。これは、ロケーションの登録に使用されたロールによって指定されている必要があります。)

これが唯一のデータロケーション許可です。この許可を持つプリンシパルは、指定された Amazon S3 ロケーションをポイントするメタデータデータベースまたはテーブルを作成できます。このロケーションは登録される必要があります。ロケーションに対するデータロケーション許可を持つプリンシパルは、子ロケーションに対するロケーション許可も持っています。

Example

以下の例は、 AWS アカウント 1111-2222-3333 のユーザー datalake_user1 に s3://products/retail に対するデータロケーション許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"ResourceArn":"arn:aws:s3:::products/retail"}}'
```

基盤となるデータのクエリや更新に DATA_LOCATION_ACCESS は必要ありません。この許可は、Data Catalog リソースの作成のみに適用されます。

データロケーション許可については、「Underlying data access control」を参照してください。

DELETE

許可	付与対象リソース	付与対象に必要な追加の許可
DELETE	TABLE	(ロケーションが登録されている場合、追加の IAM 許可は必要ありません。)

この許可を持つプリンシパルは、テーブルが指定する Amazon S3 ロケーションにある基盤となるデータの挿入、更新、および読み取りを実行できます。プリンシパルは、Lake Formation コンソールでテーブルを表示し、AWS Glue API を使用してテーブルに関する情報を取得することもできます。

Example

次の例では、`retail` AWS アカウント 1111-2222-3333 のデータベース`inventory`の テーブル`datalake_user1`に対する アクセス`DELETE`許可をユーザーに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "DELETE" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory"}}'
```

この許可は、Amazon S3 内のデータにのみ適用され、Amazon Relational Database Service (Amazon RDS) などの他のデータストア内のデータには適用されません。

DESCRIBE

許可	付与対象リソース	付与対象に必要な追加の許可
DESCRIBE	テーブルリソースリンク	glue:GetTable
	データベースリソースリンク	glue:GetDatabase
DESCRIBE	DATABASE	glue:GetDatabase
DESCRIBE	TABLE	glue:GetTable
DESCRIBE	LF-Tag	glue:GetTable glue:GetDatabase lakeformation:GetResourceLFTags lakeformation:ListLFTags

許可	付与対象リソース	付与対象に必要な追加の許可

		`lakeformation:GetLFTag`
		
		`lakeformation:SearchTablesByLFTags`
		
		`lakeformation:SearchDatabasesByLFTags`

この許可を持つプリンシパルは、指定されたデータベース、テーブル、またはリソースリンクを表示できます。これ以外の Data Catalog 許可が黙示的に付与されることはなく、データアクセス許可が黙示的に付与されることもありません。統合サービスのクエリエディタにはデータベースとテーブルが表示されますが、他の Lake Formation 許可 (SELECT など) が付与されていない限り、それらに対するクエリを実行することはできません。

例えば、データベースに対する DESCRIBE を持つユーザーは、そのデータベースとすべてのデータベースメタデータ (説明、ロケーションなど) を確認できますが、データベースにどのテーブルが含まれているかは判断できず、データベースでテーブルの削除、変更、または作成を行うことはできません。同様に、テーブルに対する DESCRIBE を持つユーザーは、テーブルとテーブルメタデータ (説明、スキーマ、ロケーションなど) を確認できますが、テーブルに対してドロップ、変更、またはクエリを実行することはできません。

以下は、DESCRIBE に関する追加のルールです。

- ユーザーがデータベース、テーブル、またはリソースリンクに対する他の Lake Formation 許可を持っている場合、DESCRIBE が黙示的に付与されます。
- ユーザーがテーブルについて列のサブセットのみに対する SELECT (partial SELECT) を持っている場合、ユーザーはこれらの列のみの表示に制限されます。
- テーブルに対する partial SELECT を持つユーザーに DESCRIBE を付与することはできません。これとは逆に、DESCRIBE が付与されているテーブルに、列の包含リストや除外リストを指定することはできません。

Example

次の例では、`retail` AWS アカウント 1111-2222-3333 のデータベース`inventory-link`のテーブルリソースリンク`datalake_user1`に対する アクセスDESCRIBE許可をユーザーに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "DESCRIBE" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory-link"}}'
```

DROP

許可	付与対象リソース	付与対象に必要な追加の許可
DROP	DATABASE	glue:DeleteDatabase
DROP	TABLE	glue:DeleteTable
DROP	LF-Tag	lakeformation:DeleteLFTag
DROP	データベースリソースリンク テーブルリソースリンク	glue:DeleteDatabase glue:DeleteTable

この許可を持つプリンシパルは、Data Catalog 内のデータベース、テーブル、またはリソースリンクをドロップできます。データベースに対する DROP を、外部のアカウントまたは組織に付与することはできません。

> **Warning**
> データベースをドロップすると、データベース内のすべてのテーブルがドロップされます。

Example

次の例では、retail AWS アカウント 1111-2222-3333 のデータベースdatalake_user1のユーザーに アクセスDROP許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "DROP" --resource '{ "Database": {"Name":"retail"}}'
```

Example

以下の例は、データベース retail にあるテーブル inventory に対する DROP をユーザー datalake_user1 に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "DROP" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory"}}'
```

Example

以下の例は、データベース retail にあるテーブルリソースリンク inventory-link に対する DROP をユーザー datalake_user1 に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "DROP" --resource '{ "Table": {"DatabaseName":"retail", "Name":"inventory-
link"}}'
```

INSERT

許可	付与対象リソース	付与対象に必要な追加の許可
INSERT	TABLE	(ロケーションが登録されている場合、追加の IAM 許可は必要ありません。)

この許可を持つプリンシパルは、テーブルが指定する Amazon S3 ロケーションにある基盤となるデータの挿入、更新、および読み取りを実行できます。プリンシパルは、Lake Formation コンソールでテーブルを表示し、AWS Glue API を使用してテーブルに関する情報を取得することもできます。

Example

次の例では、retail AWS アカウント 1111-2222-3333 のデータベースinventoryの テーブルdatalake_user1に対する アクセスINSERT許可をユーザーに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
```

```
 --permissions "INSERT" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory"}}'
```

この許可は、Amazon S3 内のデータにのみ適用され、Amazon RDS などの他のデータストア内のデータには適用されません。

SELECT

許可	付与対象リソース	付与対象に必要な追加の許可
SELECT	• TABLE	(ロケーションが登録されている場合、追加の IAM 許可は必要ありません。)

この許可を持つプリンシパルは、Data Catalog 内のテーブルを表示し、テーブルが指定するロケーションにある Amazon S3 内の基盤となるデータをクエリすることができます。プリンシパルは、Lake Formation コンソールでテーブルを表示し、AWS Glue API を使用してテーブルに関する情報を取得することができます。この許可の付与時に列フィルタリングが適用された場合、プリンシパルは、包含されている列のメタデータのみを表示でき、包含されている列からのデータのみをクエリできます。

> **Note**
>
> クエリの処理時に列フィルタリングを適用するのは、統合された分析サービスの責任です。

Example

次の例では、`retail` AWS アカウント 1111-2222-3333 のデータベース`inventory`の テーブル`datalake_user1`に対する アクセス`SELECT`許可をユーザーに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "SELECT" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory"}}'
```

この許可は、Amazon S3 内のデータにのみ適用され、Amazon RDS などの他のデータストア内のデータには適用されません。

オプションの包含リストまたは除外リストを使用して、特定の列をフィルタリング (それらへのアクセスを制限) できます。包含リストは、アクセスできる列を指定します。除外リストは、アクセスできない列を指定します。包含リストまたは除外リストがない場合は、すべてのテーブル列にアクセスできます。

glue:GetTable の結果は、呼び出し元が表示許可を持っている列のみを返します。Amazon Athena および Amazon Redshift などの統合サービスは、包含リストと除外リストに従います。

Example

以下の例は、包含リストを使用して、テーブル inventory に対する SELECT をユーザー datalake_user1 に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "SELECT"  --resource '{ "TableWithColumns": {"DatabaseName":"retail",
 "Name":"inventory", "ColumnNames": ["prodcode","location","period","withdrawals"]}}'
```

Example

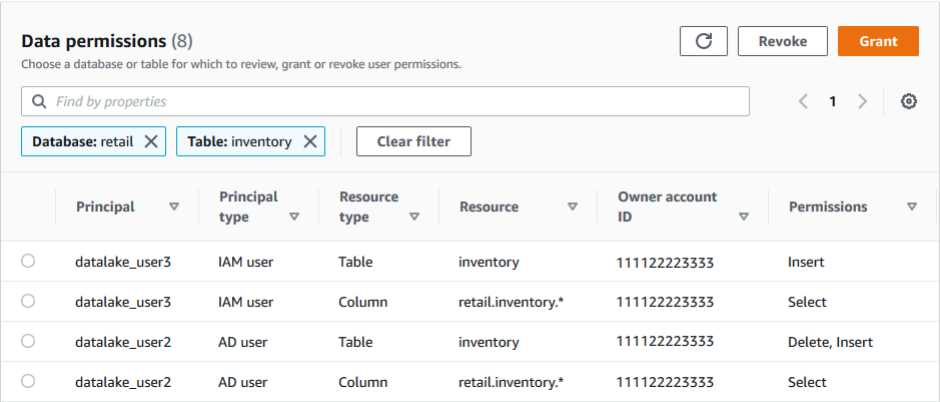
次の例は、除外リストを使用して、inventory テーブルに対する SELECT を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "SELECT"  --resource '{ "TableWithColumns": {"DatabaseName":"retail",
 "Name":"inventory", "ColumnWildcard": {"ExcludedColumnNames": ["intkey",
 "prodcode"]}}}'
```

SELECT 許可には以下の制限が適用されます。

- 列フィルタリングが適用されている場合、SELECT を付与するときに grant オプションを含めることはできません。
- パーティションキーである列に対するアクセス制御を制限することはできません。
- テーブル内の列のサブセットに対する SELECT 許可を持つプリンシパルに、そのテーブルに対する ALTER、DROP、DELETE または INSERT 許可を付与することはできません。同様に、テーブルに対する ALTER、DROP、DELETE または INSERT 許可を持つプリンシパルに、列フィルタリングを伴う SELECT 許可を付与することはできません。

SELECT 許可は常に、Lake Formation コンソールの [Data permissions] (データの許可) ページに個別の行として表示されます。以下の画像は、inventory テーブル内のすべての列に対する SELECT が、ユーザー datalake_user2 と datalake_user3 に付与されていることを示しています。



Super

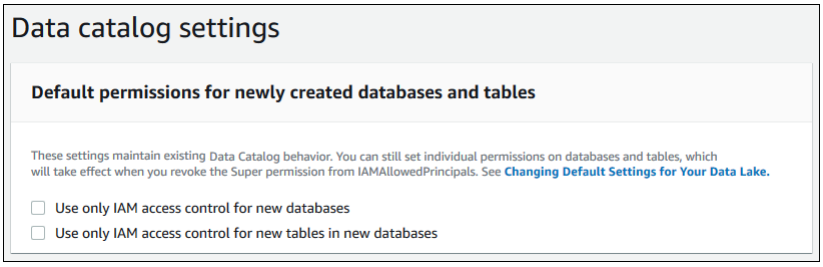
許可	付与対象リソース	付与対象に必要な追加の許可
Super	DATABASE	glue:*Database*
Super	TABLE	glue:*Table*, glue:*Partition*

この許可は、プリンシパルが、データベースまたはテーブルでサポートされているすべての Lake Formation 操作を実行できるようにします。データベースに対する Super を、外部アカウントに付与することはできません。

この許可は、他の Lake Formation 許可と共存できます。例えば、メタデータテーブルに対する Super、SELECT、および INSERT 許可を付与することができます。そうすることで、プリンシパルはテーブルに対してサポートされているすべての操作を実行できるようになります。Super を取り消すときは、SELECT と INSERT 許可が残り、プリンシパルは選択操作と挿入操作のみを実行できます。

Super は、個々のプリンシパルに付与する代わりに、グループ IAMAllowedPrincipals に付与することができます。IAMAllowedPrincipals グループは自動的に作成され、IAM ポリシーによって Data Catalog リソースへのアクセスを許可されるすべての IAM ユーザーとロールが含まれます。Data Catalog リソースに対する Super が IAMAllowedPrincipals に付与される場合、リソースへのアクセスは、実質的に IAM ポリシーのみで制御されることになります。

Lake Formation コンソールの [設定] ページにあるオプションを活用すると、新しいカタログリソースへの Super アクセス許可が自動的に IAMAllowedPrincipals に付与されるようにすることができます。



- すべての新しいデータベースに対する Super を IAMAllowedPrincipals に付与するには、[Use only IAM access control for new databases] (新しいデータベースに IAM アクセス制御のみを使用) を選択します。
- 新しいデータベース内のすべての新しいテーブルに対する Super を IAMAllowedPrincipals に付与するには、[Use only IAM access control for new databases] (新しいデータベースに IAM アクセス制御のみを使用) を選択します。

> **Note**
>
> このオプションを選択すると、[Create database] (データベースの作成) ダイアログボックスの [Use only IAM access control for new tables in this database] (このデータベース内の新しいテーブルには IAM アクセス制御のみを使用する) チェックボックスがデフォルトでオンになります。それ以上は何も行われません。IAMAllowedPrincipals への Super の付与を有効にするのは、[Create database] (データベースの作成) ダイアログボックスにあるチェックボックスです。

これらの [Settings] (設定) ページオプションは、デフォルトで有効になっています。詳細については次を参照してください:

- the section called “データレイクのデフォルト設定の変更”
- the section called “Lake Formation モデルに対する AWS Glue データの許可のアップグレード”

SUPER_USER

許可	付与対象リソース	付与対象に必要な追加の許可
Super user	Catalog	glue:GetCatalog

Super user アクセス許可は、デフォルトのデータカタログ内のカタログの特定のプリンシパルにのみ付与できます。デフォルトのカタログ、データベースやテーブルなどの他のリソースタイプ、または外部アカウントのプリンシパルに Super user アクセス許可を付与することはできません。Super user アクセス許可により、プリンシパルは、付与されたカタログ内のデータベースとテーブルに対して、サポートされているすべての Lake Formation オペレーションを実行できます。

Super user アクセス許可を使用すると、プリンシパル (被付与者) はカタログ内のリソース (カタログ、データベース、テーブル) に対して次のアクションを実行できます。

- CREATE_DATABASE、カタログに対する DESCRIBE アクセス許可。
- カタログ内のすべてのデータベースに対する DROP、ALTER、CREATE_TABLE、DESCRIBE (実質的に SUPER) アクセス許可。
- カタログ内のすべてのデータベース内のすべてのテーブルに対する DROP、ALTER、DESCRIBE、SELECT、INSERT、DELETE (実質的に SUPER) アクセス許可。
- カタログ内のカタログに対する All (実質的に SUPER) アクセス許可。
- カタログ内のすべてのカタログ、データベース、テーブルに対するアクセス許可を付与可能 (これらのアクセス許可を他のプリンシパルに付与する機能) 。

カタログリソースに対する Super user アクセス許可では、被付与者はカタログに対して ALTER アクションと DROP アクションを実行または委任することはできません。

ASSOCIATE

許可	付与対象リソース	付与対象に必要な追加の許可
ASSOCIATE	LF-Tag	glue:GetDatabase
		glue:GetTable

許可	付与対象リソース	付与対象に必要な追加の許可
		lakeformation:AddLFTagsToResource" lakeformation:RemoveLFTagsFromResource" lakeformation:GetResourceLFTags lakeformation:ListLFTags lakeformation:GetLFTag lakeformation:SearchTablesByLFTags lakeformation:SearchDatabasesByLFTags

LF タグに対してこの許可を持つプリンシパルは、LF タグを Data Catalog リソースに割り当てることができます。ASSOCIATE の付与は、DESCRIBE を黙示的に付与します。

Example

この例は、module キーを持つ LF タグに対する ASSOCIATE アクセス許可をユーザー datalake_user1 に付与します。これは、そのキーのすべての値 (アスタリスク (*) で指定) を表示して割り当てる許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1 --permissions "ASSOCIATE" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

IAM アイデンティティセンターの統合

を使用すると AWS IAM アイデンティティセンター、ID プロバイダー (IdPs) に接続し、 AWS 分析サービス全体でユーザーとグループのアクセスを一元管理できます。Okta、Ping、Microsoft Entra ID (以前は Azure Active Directory と呼ばれていました) などの ID プロバイダーを IAM アイデンティティセンターと統合すると、組織内のユーザーは、シングルサインオンエクスペリエンスを使用してデータにアクセスできるようになります。IAM アイデンティティセンターは、追加のサードパーティ ID プロバイダーとの接続もサポートしています。

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「サポートされている ID プロバイダー」を参照してください。

を IAM Identity Center で有効なアプリケーション AWS Lake Formation として設定でき、データレイク管理者は AWS Glue Data Catalog リソースの承認されたユーザーとグループにきめ細かなアクセス許可を付与できます。

組織のユーザーは、組織の ID プロバイダーを使用してアイデンティティセンター対応アプリケーションにサインインし、Lake Formation 許可を適用してデータセットにクエリを実行できます。この統合により、複数の IAM ロールを作成することなく、 AWS サービスへのアクセスを管理できます。

信頼できる ID 伝達は、接続された の管理者がサービスデータへのアクセスを許可および監査するために AWS のサービス 使用できる AWS IAM アイデンティティセンター 機能です。このデータへのアクセスは、グループの関連付けなどのユーザー属性に基づいています。信頼できる ID 伝達を設定するには、接続されている の管理者 AWS のサービス と IAM アイデンティティセンターの管理者間のコラボレーションが必要です。詳細については、「Prerequisites and considerations」を参照してください。

制限事項については、「IAM アイデンティティセンター 統合の制限事項」を参照してください。

トピック

- IAM アイデンティティセンターを Lake Formation と統合するための前提条件
- Lake Formation と IAM アイデンティティセンターとの接続
- IAM アイデンティティセンター統合の更新
- IAM アイデンティティセンターとの Lake Formation 統合の削除
- ユーザーおよびグループへのアクセス許可の付与
- CloudTrail ログへの IAM アイデンティティセンターのユーザーコンテキストの追加

IAM アイデンティティセンターを Lake Formation と統合するための前提条件

IAM アイデンティティセンターを Lake Formation と統合するための前提条件は次のとおりです。

1. IAM アイデンティティセンターを有効にする — IAM アイデンティティセンターを有効にすることは、認証と ID の伝播をサポートするための前提条件です。
2. ID ソースを選択する — IAM アイデンティティセンターを有効にしたら、ユーザーとグループを管理する ID プロバイダーが必要になります。組み込まれているアイデンティティセンターディレクトリをアイデンティティソースとして使用することも、Microsoft Entra ID や Okta などの外部 IdP を使用することもできます。

   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「ID ソースの管理」および「外部 ID プロバイダーへの接続」を参照してください。
3. IAM ロールを作成する — IAM アイデンティティセンター接続を作成するロールには、以下のインラインポリシーのように、Lake Formation と IAM アイデンティティセンターでアプリケーション設定を作成および変更するアクセス許可が必要です。

   IAM のベストプラクティスに従ってアクセス許可を追加する必要があります。特定のアクセス許可については、以降の手順で詳しく説明します。詳細については、「IAM アイデンティティセンターの開始方法」を参照してください。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:CreateLakeFormationIdentityCenterConfiguration",
                "sso:CreateApplication",
                "sso:PutApplicationAssignmentConfiguration",
                "sso:PutApplicationAuthenticationMethod",
                "sso:PutApplicationGrant",
                "sso:PutApplicationAccessScope"
            ],
            "Resource": [
                "*"
            ]
```

```
        }
    ]
}
```

Data Catalog リソースを外部 AWS アカウント または組織と共有する場合は、リソース共有を作成するための AWS Resource Access Manager (AWS RAM) アクセス許可が必要です。リソースの共有に必要なアクセス許可の詳細については、「クロスアカウントデータ共有の一般的な要件」を参照してください。

以下のインラインポリシーには、Lake Formation と IAM アイデンティティセンターの統合のプロパティを表示、更新、削除するために必要な特定の権限が含まれています。

- 以下のインラインポリシーを使用して、IAM ロールで Lake Formation と IAM アイデンティティセンターの統合を表示できるようにします。

  JSON

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

 "lakeformation:DescribeLakeFormationIdentityCenterConfiguration",
                "sso:DescribeApplication"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

- 以下のインラインポリシーを使用して、IAM ロールで Lake Formation と IAM アイデンティティセンターの統合を更新できるようにします。このポリシーには、外部アカウントとリソースを共有するために必要なオプションのアクセス許可も含まれています。

  JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:UpdateLakeFormationIdentityCenterConfiguration",

 "lakeformation:DescribeLakeFormationIdentityCenterConfiguration",
                "sso:DescribeApplication",
                "sso:UpdateApplication"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

- 以下のインラインポリシーを使用して、IAM ロールで Lake Formation と IAM アイデンティティセンターの統合を削除できるようにします。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:DeleteLakeFormationIdentityCenterConfiguration",
                "sso:DeleteApplication"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

- IAM アイデンティティセンターのユーザーとグループにデータレイクのアクセス許可を付与または取り消すために必要な IAM アクセス許可については、「Lake Formation 許可の付与と取り消しに必要な IAM 許可」を参照してください。

アクセス許可の説明

- lakeformation:CreateLakeFormationIdentityCenterConfiguration – Lake Formation IdC 設定を作成します。
- lakeformation:DescribeLakeFormationIdentityCenterConfiguration – 既存の IdC 設定について説明します。
- lakeformation:DeleteLakeFormationIdentityCenterConfiguration – 既存のLake Formation IdC 設定を削除できます。
- lakeformation:UpdateLakeFormationIdentityCenterConfiguration – 既存のLake Formation 設定を変更するために使用されます。
- sso:CreateApplication – IAM アイデンティティセンターのアプリケーションを作成するために使用されます。
- sso:DeleteApplication – IAM アイデンティティセンターのアプリケーションを削除するために使用されます。
- sso:UpdateApplication – IAM アイデンティティセンターのアプリケーションの更新に使用されます。
- sso:PutApplicationGrant – 信頼できるトークン発行者の情報を変更するために使用されます。
- sso:PutApplicationAuthenticationMethod – Lake Formation 認証アクセスを許可します。
- sso:GetApplicationGrant – 信頼できるトークン発行者の情報を一覧表示するために使用されます。
- sso:DeleteApplicationGrant – 信頼できるトークン発行者の情報を削除します。
- sso:PutApplicationAccessScope – アプリケーションの IAM アイデンティティセンターアクセススコープの承認済みターゲットのリストを追加または更新します。
- sso:PutApplicationAssignmentConfiguration – ユーザーがアプリケーションにアクセスする方法を設定するために使用されます。

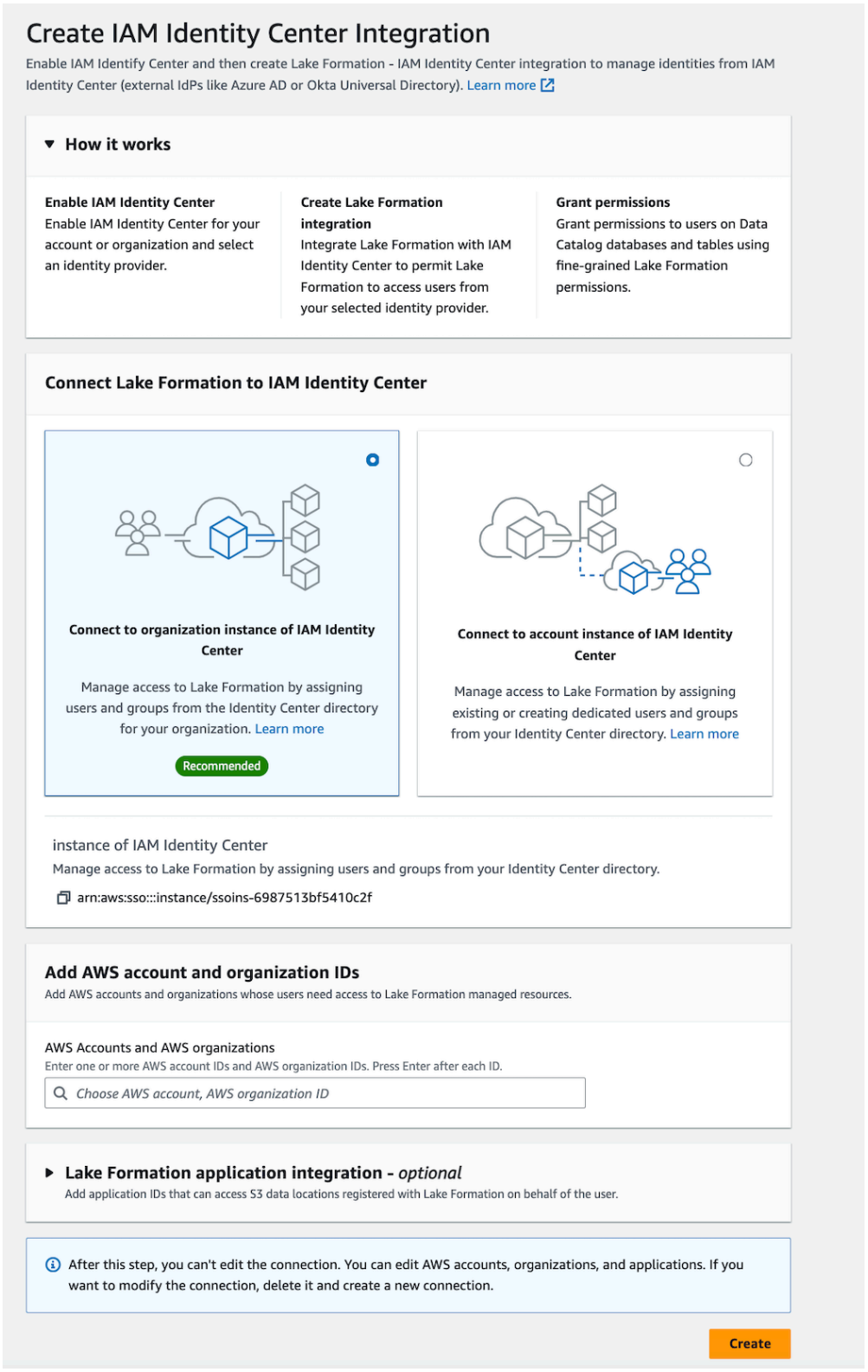
Lake Formation と IAM アイデンティティセンターとの接続

IAM アイデンティティセンターを使用して ID を管理し、Lake Formation を使用してデータカタログリソースへのアクセスを許可する前に、次の手順を完了する必要があります。Lake Formation コンソールまたは AWS CLIを使用して IAM アイデンティティセンター統合を作成できます。

AWS マネジメントコンソール

Lake Formation を IAM アイデンティティセンターと接続するには

1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation 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、[IAM アイデンティティセンターの統合] を選択します。



3. (オプション) 1 つ以上の AWS アカウント IDs、組織 IDs、組織単位 IDs を入力して、外部アカウントが Data Catalog リソースにアクセスできるようにします。IAM アイデンティティセンターのユーザーまたはグループが Lake Formation で管理されているデータカタログリソー

スにアクセスしようとすると、Lake Formation は IAM ロールを引き受けてメタデータアクセスを認可します。IAM ロールがリソースポリシーと AWS Glue リソース AWS RAM 共有を持たない外部アカウントに属している場合、IAM アイデンティティセンターのユーザーとグループは、Lake Formation アクセス許可があってもリソースにアクセスできません。

Lake Formation は AWS Resource Access Manager 、 (AWS RAM) サービスを使用して外部アカウントおよび組織とリソースを共有します。 は、リソース共有を承認または拒否するための招待を被付与者アカウント AWS RAM に送信します。

詳細については、「からのリソース共有の招待の承諾 AWS RAM」を参照してください。

> **Note**
>
> Lake Formation は、データカタログリソースへのアクセスのために、外部アカウントの IAM ロールが IAM アイデンティティセンターのユーザーとグループに代わってキャリアロールとして動作することを許可しますが、アクセス許可を付与できるのは、所有アカウント内のデータカタログリソースに対してだけです。外部アカウント内のデータカタログリソースに対するアクセス許可を IAM アイデンティティセンターのユーザーとグループに付与しようとすると、Lake Formation から「Cross-account grants are not supported for the principal」というエラーがスローされます。

4. (オプション) [Lake Formation 統合の作成] 画面で、Lake Formation に登録された Amazon S3 ロケーションにあるデータにアクセスできるサードパーティアプリケーションの ARN を指定します。Lake Formation は、有効なアクセス許可に基づいて、スコープダウンされた一時的な認証情報を AWS STS トークンの形式で登録された Amazon S3 ロケーションに提供し、承認されたアプリケーションがユーザーに代わってデータにアクセスできるようにします。
5. (オプション) Lake Formation 統合の作成画面で、Trusted Identity Propagation の Amazon Redshift Connect チェックボックスをオンにして、IDC を介した Amazon Redshift フェデレーティッドアクセス許可の検出を有効にします。Lake Formation は、有効なアクセス許可に基づいて ID をダウンストリームに伝播するため、承認されたアプリケーションはユーザーに代わってデータにアクセスできます。
6. [送信] を選択します。

Lake Formation 管理者が手順を完了して統合を作成すると、IAM アイデンティティセンターのプロパティが Lake Formation コンソールに表示されます。上記のタスクを完了すると、Lake Formation は IAM アイデンティティセンター対応アプリケーションになります。コンソールのプロパティには統合ステータスが含まれます。統合が完了すると、統合ステー

タスに Success と表示されます。このステータスは IAM アイデンティティセンターの設定が完了したかどうかを示します。

AWS CLI

- 次の例は、IAM アイデンティティセンターとの Lake Formation 統合を作成する方法を示しています。アプリケーションの Status (ENABLED、DISABLED) を指定することもできます。

```
aws lakeformation create-lake-formation-identity-center-configuration \
    --catalog-id <123456789012> \
    --instance-arn <arn:aws:sso:::instance/ssoins-112111f12ca1122p> \
    --share-recipients '[{"DataLakePrincipalIdentifier": "<123456789012>"},
                         {"DataLakePrincipalIdentifier": "<555555555555>"}]' \
    --external-filtering '{"AuthorizedTargets": ["<app arn1>", "<app arn2>"],
 "Status": "ENABLED"}'
```

- 次の例は、IAM アイデンティティセンターとの Lake Formation 統合を表示する方法を示しています。

```
aws lakeformation describe-lake-formation-identity-center-configuration
 --catalog-id <123456789012>
```

- 次の例は、Redshift:Connect認可を有効にする方法を示しています。認可は ENABLED または DISABLED にすることができます。

```
aws lakeformation  create-lake-formation-identity-center-configuration \
--instance-arn <arn:aws:sso:::instance/ssoins-112111f12ca1122p> \
--service-integrations '[{
  "Redshift": [{
    "RedshiftConnect": {
      "Authorization": "ENABLED"
    }
  }]
}]'
```

- describe-lake-formation-identity-center-configuration コマンドを使用して、レイクフォーメーションアイデンティティセンターアプリケーションを記述します。Redshift:Connectサービス統合は、クロスサービスおよびクラスター間の IdC ID の伝播に不可欠です。

```
aws lakeformation describe-lake-formation-identity-center-configuration --catalog-
id <123456789012>
```

レスポンス:

```
{
    "CatalogId": "CATALOG ID",
    "InstanceArn": "INSTANCE ARN",
    "ApplicationArn": "APPLICATION ARN",
    "ShareRecipients": [],
    "ServiceIntegrations": [
        {
            "Redshift": [
                {
                    "RedshiftConnect": {
                        "Authorization": "ENABLED"
                    }
                }
            ]
        }
    ]
}
```

複数の で IAM Identity Center を使用する AWS リージョン

Lake Formation は、複数の で IAM Identity Center をサポートしています AWS リージョン。IAM アイデンティティセンターをプライマリリージョンから追加のリージョン AWS リージョン に拡張して、ユーザーと信頼性に近接することでパフォーマンスを向上させることができます。IAM アイデンティティセンターに新しいリージョンが追加されると、プライマリリージョンの ID をレプリケートすることなく、新しいリージョンに Lake Formation アイデンティティセンターアプリケーションを作成できます。複数のリージョンで IAM アイデンティティセンターの使用を開始する方法の詳細については、IAM アイデンティティセンターユーザーガイドの「マルチリージョン IAM アイデンティティセンター」を参照してください。

IAM アイデンティティセンター統合の更新

接続を作成したら、IAM アイデンティティセンター統合のサードパーティのアプリケーションを追加して Lake Formation と統合し、ユーザーに代わって Amazon S3 データにアクセスできるようになります。既存のアプリケーションを IAM アイデンティティセンター統合から削除することもでき

ます。Lake Formation コンソール、および UpdateLakeFormationIdentityCenterConfiguration オペレーションを使用して AWS CLI、アプリケーションを追加または削除できます。

> **Note**
>
> IAM アイデンティティセンター統合を作成した後は、インスタンスの ARN を更新することはできません。

AWS マネジメントコンソール

Lake Formation との既存の IAM アイデンティティセンターの接続を更新するには

1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation 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、[IAM アイデンティティセンターの統合] を選択します。
3. [IAM アイデンティティセンターの統合] ページで [追加] を選択します。
4. 1 つ以上の AWS アカウント IDs、組織 IDs、組織単位 IDs を入力して、外部アカウントが Data Catalog リソースにアクセスできるようにします。
5. [アプリケーションの追加] 画面で、Lake Formation と統合するサードパーティアプリケーションのアプリケーション ID を入力します。
6. [追加] を選択します。
7. (Optioanlly) IAM Identity Center 統合ページで、Amazon Redshift 接続の信頼できる ID 伝達を有効にするか、無効にすることができます。Lake Formation は、有効なアクセス許可に基づいて ID をダウンストリームに伝播するため、承認されたアプリケーションはユーザーに代わってデータにアクセスできます。

AWS CLI

次の AWS CLI コマンドを実行して、IAM Identity Center 統合用のサードパーティーアプリケーションを追加または削除できます。外部フィルタリングステータスを ENABLED に設定すると、IAM アイデンティティセンターで、Lake Formation によって管理されるデータにアクセスするためのサードパーティのアプリケーションの ID 管理を提供できるようになります。また、アプリケーションステータスを設定することで、IAM アイデンティティセンター統合を有効または無効にすることもできます。

```
aws lakeformation update-lake-formation-identity-center-configuration \
```

```
 --external-filtering '{"AuthorizedTargets": ["<app arn1>", "<app arn2>"], "Status":
 "ENABLED"}'\
 --share-recipients '[{"DataLakePrincipalIdentifier": "<444455556666>"}
                     {"DataLakePrincipalIdentifier": "<777788889999>"}]' \
 --application-status ENABLED
```

既存の LF IDC アプリケーションがあり、Redshift:Connect認可を追加する場合は、以下を使用して Lake Formation IDC アプリケーションを更新できます。認可は ENABLED または DISABLED にすることができます。

```
aws lakeformation update-lake-formation-identity-center-configuration \
--service-integrations '[{

  "Redshift": [{
    "RedshiftConnect": {
      "Authorization": "ENABLED"
    }
  }]
}]'
```

IAM アイデンティティセンターとの Lake Formation 統合の削除

既存の IAM Identity Center 統合を削除する場合は、Lake Formation コンソール AWS CLI、または DeleteLakeFormationIdentityCenterConfiguration オペレーションを使用して削除できます。

AWS マネジメントコンソール

Lake Formation との既存の IAM アイデンティティセンターの接続を削除するには

1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation 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、[IAM アイデンティティセンターの統合] を選択します。
3. [IAM アイデンティティセンターの統合] ページで [削除] を選択します。
4. [統合の確認] 画面でアクションを確認し、[削除] を選択します。

AWS CLI

IAM Identity Center 統合を削除するには、次の AWS CLI コマンドを実行します。

```
aws lakeformation delete-lake-formation-identity-center-configuration \
    --catalog-id <123456789012>
```

ユーザーおよびグループへのアクセス許可の付与

データレイク管理者は、データカタログリソース (データベース、テーブル、ビュー) について IAM アイデンティティセンターのユーザーとグループにアクセス許可を付与できます。これにより、データに簡単にアクセスできるようになります。データレイクのアクセス許可を付与または取り消すには、付与者に次の IAM アイデンティティセンターアクションに対するアクセス許可が必要です。

- DescribeUser
- DescribeGroup
- DescribeInstance

許可は、Lake Formation コンソール、API、または AWS CLIを使用して付与することができます。

許可の付与の詳細については、「the section called “データアクセス許可の付与”」を参照してください。

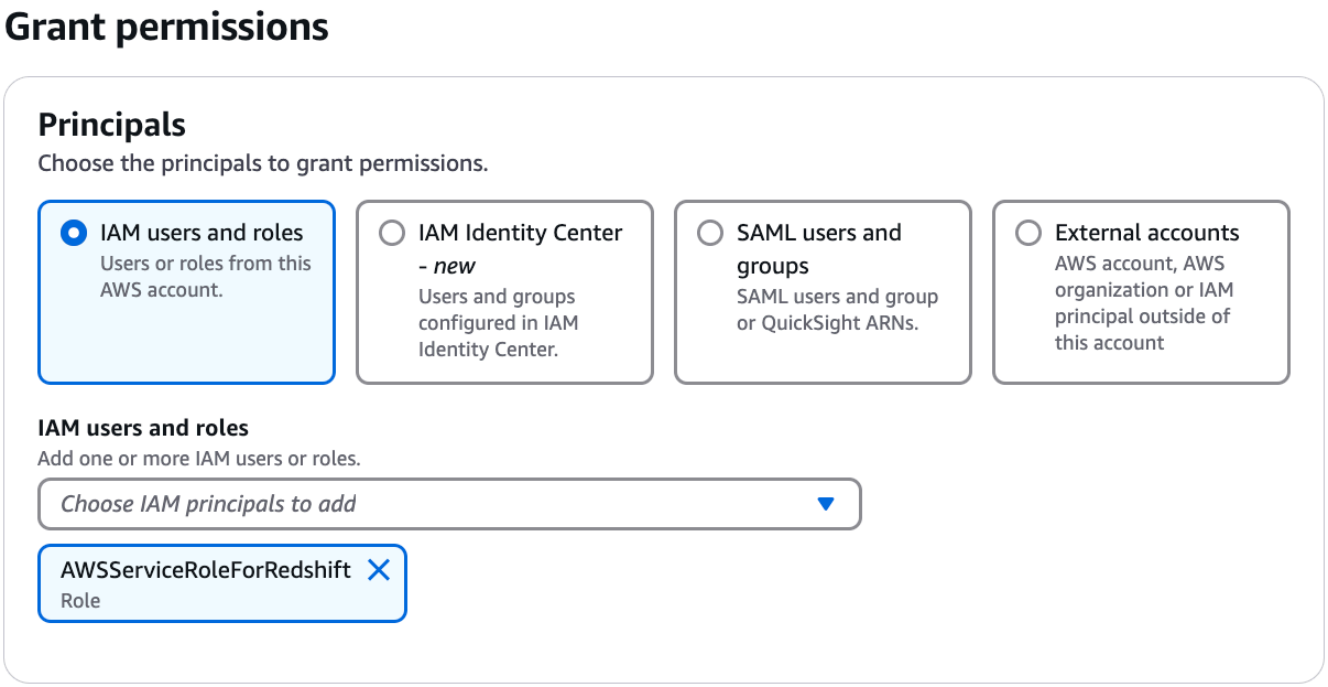
> **Note**
>
> アクセス許可は、アカウント内のリソースに対してのみ付与できます。共有されているリソースのユーザーとグループに許可をカスケードするには、 AWS RAM リソース共有を使用する必要があります。

AWS マネジメントコンソール

ユーザーおよびグループにアクセス許可を付与するには

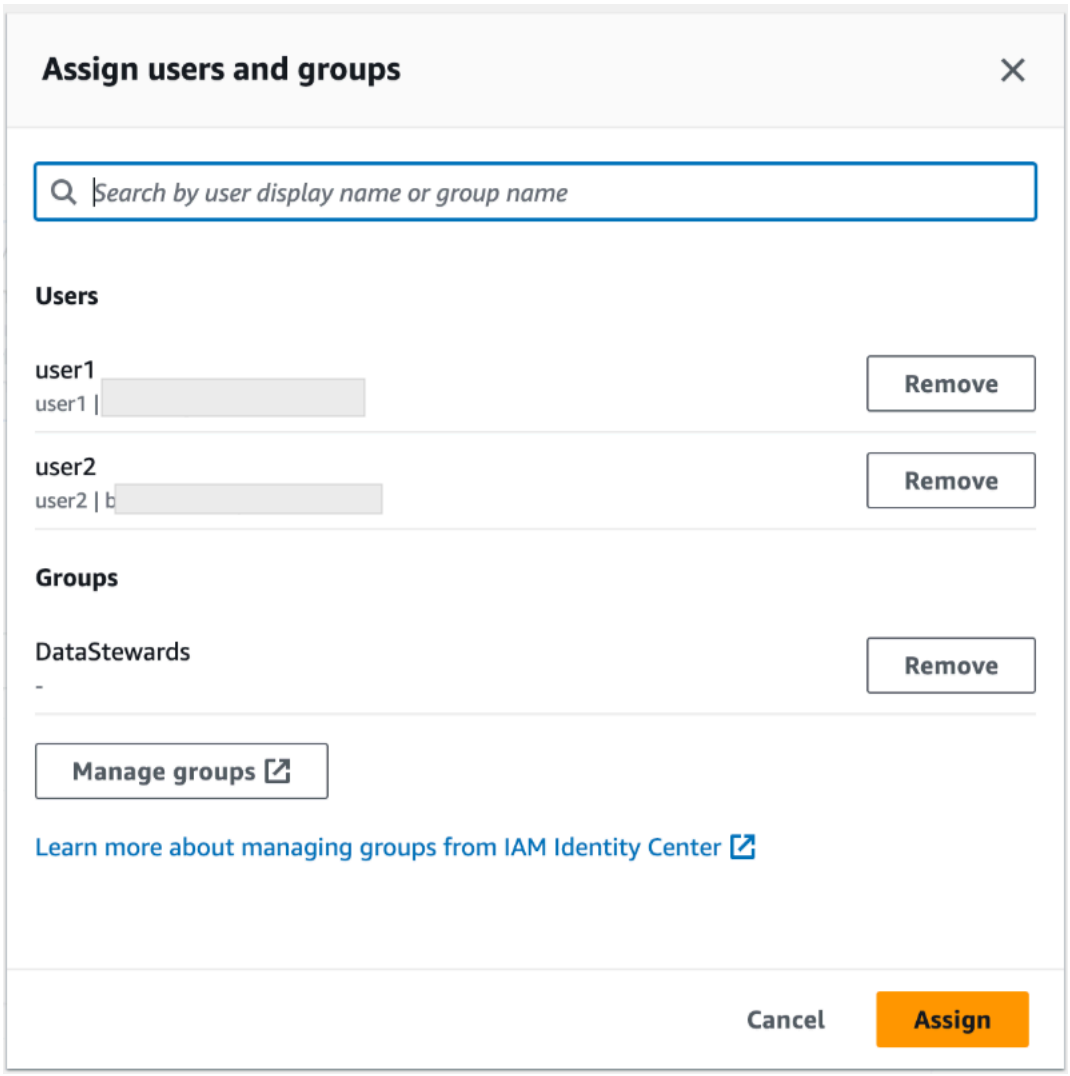
1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールを開きます。
2. Lake Formation コンソールの [許可] で [データレイクのアクセス許可] を選択します。
3. [付与] を選択します。
4. [データレイクのアクセス許可の付与] ページで、[IAM アイデンティティセンター] のユーザーとグループを選択します。

5. [追加] を選択して、許可を付与するユーザーとグループを選択します。



6. [ユーザーとグループの割り当て] 画面で、許可を付与するユーザーやグループを選択します。

   [割り当て] を選択します。



7. 次に、許可を付与する方法を選択します。

   名前付きリソース方式を使用して許可を付与する手順については、「名前付きリソース方式を使用したデータアクセス許可の付与」を参照してください。

   LF タグを使用して許可を付与する手順については、「LF-TBAC 方式を使用したデータレイク許可の付与」を参照してください。

8. 許可を付与するデータカタログリソースを選択します。
9. 付与するデータカタログのアクセス許可を選択します。
10. [付与] を選択します。

AWS CLI

次の例は、テーブルに対する SELECT 許可を IAM アイデンティティセンターユーザーに付与する方法を示しています。

```
aws lakeformation grant-permissions \
--principal DataLakePrincipalIdentifier=arn:aws:identitystore:::user/<UserId> \
--permissions "SELECT" \
--resource '{ "Table": { "DatabaseName": "retail", "TableWildcard": {} } }'
```

IAM アイデンティティセンターから UserId を取得するには、「IAM アイデンティティセンター API リファレンス」で GetUserId オペレーションを参照してください。

CloudTrail ログへの IAM アイデンティティセンターのユーザーコンテキストの追加

Lake Formation では、認証情報の供給を使用して Amazon S3 データへの一時的なアクセスを提供します。統合された分析サービスに IAM アイデンティティセンターユーザーがクエリを送信した場合、デフォルトで CloudTrail ログには、サービスが短期間のアクセスを提供するために引き受けた IAM ロールのみが記録されます。ユーザー定義ロールを使用して Amazon S3 データロケーションを Lake Formation に登録すると、CloudTrail イベントに IAM アイデンティティセンターユーザーのコンテキストを含めるようにオプトインし、リソースにアクセスするユーザーを追跡できます。

> **Important**
>
> オブジェクトレベルの Amazon S3 API リクエストを CloudTrail に含めるには、Amazon S3 バケットとオブジェクトの CloudTrail イベントログを有効にする必要があります。詳細については、「Amazon S3 ユーザーガイド」の「S3 バケットとオブジェクトの CloudTrail イベントログ記録の有効化」を参照してください。

ユーザー定義ロールを使用して登録されたデータレイクロケーションで認証情報供給の監査を有効にするには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインします。

2. 左側のナビゲーションで、[管理] を展開し、[データカタログの設定] を選択します。
3. [拡張監査] で、[提供されたコンテキストを伝播] を選択します。
4. [保存] を選択します。

拡張監査オプションは、PutDataLakeSettings オペレーションで Parameters 属性を設定することでも有効にできます。デフォルトでは、SET_CONTEXT" パラメータの値は「true」に設定されます。

```
{
    "DataLakeSettings": {
        "Parameters": {"SET_CONTEXT": "true"},
    }
}
```

以下は、拡張監査オプションを有効にした場合の CloudTrail イベントからの抜粋です。このログには、IAM アイデンティティセンターユーザーのセッションコンテキストと、Amazon S3 データロケーションにアクセスするために Lake Formation によって引き受けられたユーザー定義の IAM ロールの両方が含まれています。以下の抜粋の onBehalfOf パラメーターを参照してください。

```
{
         "eventVersion":"1.09",
         "userIdentity":{
            "type":"AssumedRole",
            "principalId":"AROAW7F7MOX4OYE6FLIFN:access-grants-
e653760c-4e8b-44fd-94d9-309e035b75ab",
            "arn":"arn:aws:sts::123456789012:assumed-role/accessGrantsTestRole/access-
grants-e653760c-4e8b-44fd-94d9-309e035b75ab",
            "accountId":"123456789012",
            "accessKeyId":"ASIAW7F7MOX4CQLD4JIZN",
            "sessionContext":{
               "sessionIssuer":{
                  "type":"Role",
                  "principalId":"AROAW7F7MOX4OYE6FLIFN",
                  "arn":"arn:aws:iam::123456789012:role/accessGrantsTestRole",
                  "accountId":"123456789012",
                  "userName":"accessGrantsTestRole"
               },
               "attributes":{
                  "creationDate":"2023-08-09T17:24:02Z",
                  "mfaAuthenticated":"false"
```

```
            }
         },
         "onBehalfOf":{
             "userId": "<identityStoreUserId>",
             "identityStoreArn": "arn:aws:identitystore::<restOfIdentityStoreArn>"
         }
      },
      "eventTime":"2023-08-09T17:25:43Z",
      "eventSource":"s3.amazonaws.com",
      "eventName":"GetObject",
....
```

データレイクへの Amazon S3 ロケーションの追加

データロケーションをデータレイクのストレージとして追加するには、そのロケーション (データレイクの場所) を に登録します AWS Lake Formation。その後、Lake Formation アクセス許可を使用して、この場所を指す AWS Glue Data Catalog オブジェクトと、その場所の基盤となるデータへのきめ細かなアクセスコントロールを行うことができます。

また、Lake Formation では、ハイブリッドアクセスモードでデータロケーションを登録でき、Data Catalog 内のデータベースとテーブルに対して Lake Formation 許可を選択的に有効にできる柔軟性があります。ハイブリッドアクセスモードでは、増分パスにより、他の既存のユーザーやワークロードのアクセス許可ポリシーを中断することなく、特定のユーザーのセットに Lake Formation アクセス許可を設定できます。

ハイブリッドアクセスモードアクセスの詳細については、「ハイブリッドアクセスモード」を参照してください。

ロケーションを登録すると、その Amazon S3 パスと、そのパスの下にあるすべてのフォルダが登録されます。

例えば、以下のような Amazon S3 パス組織があるとします。

`/mybucket/accounting/sales/`

`S3://mybucket/accounting` を登録すると、`sales` フォルダも登録され、Lake Formation の管理下に置かれます。

ロケーションの登録に関する詳細については、「Underlying data access control」を参照してください。

> **Note**
>
> Lake Formation 許可は、構造化データ (行と列がある表にまとめられたデータ) が推奨されます。データにオブジェクトベースの非構造化データが含まれている場合は、Amazon S3 Access Grants を使用してデータアクセスを管理することを検討してください。

トピック

- ロケーションの登録に使用されるロールの要件
- Amazon S3 ロケーションの登録
- 暗号化された Amazon S3 ロケーションの登録
- 別の AWS アカウントでの Amazon S3 ロケーションの登録
- AWS アカウント間で暗号化された Amazon S3 の場所を登録する
- Amazon S3 ロケーションの登録解除
- Amazon S3 ロケーションへのアクセス

ロケーションの登録に使用されるロールの要件

Amazon Simple Storage Service AWS Identity and Access Management (Amazon S3) の場所を登録するときは、 (IAM) ロールを指定する必要があります。 はその場所のデータにアクセスするときにそのロールを AWS Lake Formation 引き受けます。

ロケーションは、以下のロールタイプのいずれかを使用して登録できます。

- Lake Formation サービスリンクロール。このロールは、ロケーションに対する必要な許可を付与します。このロールの使用は、ロケーションを登録する最もシンプルな方法です。詳細については、「Lake Formation のサービスリンクロールの使用」および「サービスにリンクされたロールの制限」を参照してください。
- ユーザー定義のロール。ユーザー定義のロールは、サービスリンクロールが提供する許可よりも多くの許可を付与する必要があるときに使用します。

  以下の状況では、ユーザー定義のロールを使用する必要があります。

  - 別のアカウントにあるロケーションを登録する場合。

詳細については、「the section called “別の AWS アカウントでの Amazon S3 ロケーションの登録”」および「the section called “AWS アカウント間で暗号化された Amazon S3 の場所を登録する”」を参照してください。

- AWS マネージド CMK (aws/s3) を使用して Amazon S3 の場所を暗号化した場合。

  詳細については、「暗号化された Amazon S3 ロケーションの登録」を参照してください。

- Amazon EMR を使用してロケーションにアクセスする予定の場合。

  サービスリンクロールを使用してロケーションをすでに登録しており、Amazon EMR を使用したロケーションへのアクセスを開始したいという場合は、ロケーションの登録を解除してから、ユーザー定義のロールを使用して再度登録する必要があります。詳細については、「the section called “Amazon S3 ロケーションの登録解除”」を参照してください。

Lake Formation のサービスリンクロールの使用

AWS Lake Formation 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。サービスリンクロールは、Lake Formation に直接リンクされた特殊なタイプの IAM ロールです。サービスにリンクされたロールは Lake Formation によって事前定義されており、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。

ロールを作成して必要な許可を手動で追加する必要がないため、サービスリンクロールは Lake Formation のセットアップを容易にします。サービスリンクロールの許可は Lake Formation が定義し、別途定義されている場合を除いて、Lake Formation のみがそのロールを引き受けることができます。定義された許可には信頼ポリシーと許可ポリシーが含まれ、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。

このサービスリンクロールは、ロールの引き受けについて以下のサービスを信頼します。

- `lakeformation.amazonaws.com`

アカウント A のサービスリンクロールを使用して、アカウント B が所有する Amazon S3 ロケーションを登録する場合は、アカウント B の Amazon S3 バケットポリシー (リソースベースのポリシー) で、アカウント A のサービスリンクロールにアクセス許可を付与する必要があります。

サービスリンクロールを使用してデータの場所を登録する方法については、「サービスにリンクされたロールの制限」を参照してください。

> **Note**
>
> サービスリンクロールは、サービスコントロールポリシー (SCP) の影響を受けません。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー (SCP)」を参照してください。

Lake Formation のサービスリンクロールの許可

Lake Formation は、AWSServiceRoleForLakeFormationDataAccess という名前のサービスリンクロールを使用します。このロールは、Lake Formation 統合サービス (など) が登録済みロケーションにアクセスできるようにする Amazon Simple Storage Service (Amazon S3 Amazon Athena) アクセス許可のセットを提供します。データレイクロケーションを登録するときは、そのロケーションに対する必要な Amazon S3 読み取り/書き込み許可を持つロールを指定する必要があります。ユーザーは、必要な Amazon S3 許可を持つロールを作成する代わりに、このサービスリンクロールを使用することができます。

パスを登録するためのロールとしてサービスリンクロールを初めて指定すると、ユーザーに代わってサービスリンクロールと新しい IAM ポリシーが作成されます。Lake Formation がインラインポリシーにそのパスを追加し、ポリシーをサービスリンクロールにアタッチします。サービスリンクロールに後続のパスを登録すると、Lake Formation がそのパスを既存のポリシーに追加します。

データレイク管理者としてサインインしているときに、データレイクロケーションを登録します。次に、IAM コンソールで AWSServiceRoleForLakeFormationDataAccess ロールを検索し、アタッチされたポリシーを確認します。

例えば、s3://my-kinesis-test/logs のロケーションを登録すると、Lake Formation が以下のインラインポリシーを作成し、AWSServiceRoleForLakeFormationDataAccess にアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LakeFormationDataAccessPermissionsForS3",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": [
                "arn:aws:s3:::my-kinesis-test/logs/*"
            ]
        },
        {
            "Sid": "LakeFormationDataAccessPermissionsForS3ListBucket",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::my-kinesis-test"
            ]
        }
    ]
}
```

Lake Formation のサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。Amazon S3 ロケーションを AWS マネジメントコンソール、、 AWS CLIまたは AWS API で Lake Formation に登録すると、Lake Formation によってサービスにリンクされたロールが作成されます。

> **Important**
>
> このサービスリンクロールは、このロールでサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。詳細については、IAM アカウントに新しいロールが表示される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た後で再度作成する必要が生じた場合は、同じ手順でアカウントにロールを再作成できます。Amazon S3 ロケーションを Lake Formation に登録すると、Lake Formation によってサービスリンクロールが再度作成されます。

IAM コンソールを使用して、Lake Formation ユースケースでサービスリンクロール
を作成することもできます。 AWS CLI または AWS API で、サービス名を使用し
てlakeformation.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。

Lake Formation のサービスリンクロールの編集

Lake Formation では、AWSServiceRoleForLakeFormationDataAccess サービスリンク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。

Lake Formation のサービスリンクロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、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。

> **Note**
>
> リソースを削除しようとしたときに Lake Formation サービスでロールが使用されていると、削除に失敗することがあります。失敗した場合は数分待ってから操作を再試行してください。

Lake Formation で使用されている Lake Formation リソースを削除するには

- サービスリンクロールを使用して Amazon S3 ロケーションを Lake Formation に登録した場合は、サービスリンクロールを削除する前に、そのロケーションを登録解除し、カスタムロールを使用して再登録する必要があります。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、 AWS CLI、または AWS API を使用し
て、AWSServiceRoleForLakeFormationDataAccess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。

以下は、ユーザー定義のロールの要件です。

- 新しいロールを作成するときは、IAM コンソールの [ロールの作成] ページで [AWS のサービス] を選択してから、[ユースケースの選択] で [Lake Formation] を選択します。

  別のパスを使用してロールを作成する場合は、そのロールに `lakeformation.amazonaws.com` との信頼関係があることを確認します。詳細については、「ロール信頼ポリシーの更新 (コンソール)」を参照してください。
- ロールには、ロケーションに対する Amazon S3 の読み取り/書き込み許可を付与するインラインポリシーが必要です。以下は典型的なポリシーです。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::awsexamplebucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::awsexamplebucket"
            ]
        }
    ]
}
```

- Lake Formation サービスでロールを引き受け、統合された分析エンジンに一時的な認証情報を提供できるようにするには、IAM ロールに次の信頼ポリシーを追加します。

  CloudTrail ログに IAM アイデンティティセンターのユーザーコンテキストを含めるには、信頼ポリシーに `sts:SetContext` アクションのアクセス許可が必要です。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DataCatalogViewDefinerAssumeRole1",
            "Effect": "Allow",
            "Principal": {
               "Service": [
                    "lakeformation.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ]
        }
    ]
}
```

- ロケーションを登録するデータレイク管理者は、ロールに対する `iam:PassRole` 許可を持っている必要があります。

  以下は、この許可を付与するインラインポリシーです。*<account-id>* を有効な AWS アカウント番号に置き換え、*<role-name>* をロールの名前に置き換えます。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PassRolePermissions",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
```

```
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/<role-name>"
            ]
        }
    ]
}
```

- Lake Formation が CloudWatch Logs にログを追加し、メトリクスを発行できるようにするには、以下のインラインポリシーを追加します。

> **Note**
>
> CloudWatch Logs への書き込みには料金が発生します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Sid1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": [
                 "arn:aws:logs:us-east-1:111122223333:log-group:/aws-
lakeformation-acceleration/*",
                 "arn:aws:logs:us-east-1:111122223333:log-group:/aws-
lakeformation-acceleration/*:log-stream:*"
            ]
        }
    ]
}
```

Amazon S3 ロケーションの登録

Amazon Simple Storage Service AWS Identity and Access Management (Amazon S3) ロケーションを登録するときは、 (IAM) ロールを指定する必要があります。Lake Formation は、その場所のデータにアクセスする統合 AWS サービスに一時的な認証情報を付与するときに、そのロールを引き受けます。

> **Important**
>
> [Requester pays] (リクエスタ支払い) が有効になっている Amazon S3 バケットの登録は避けてください。Lake Formation に登録されたバケットの場合、バケットの登録に使用されるロールは常にリクエスト元であると見なされます。バケットが別の AWS アカウントからアクセスされた場合、ロールがバケット所有者と同じアカウントに属している場合、バケット所有者はデータアクセスに対して課金されます。

AWS Lake Formation コンソール、Lake Formation API、または AWS Command Line Interface (AWS CLI) を使用して Amazon S3 の場所を登録できます。

[開始する前に]

「ロケーションの登録に使用されるロールの要件」を確認してください。

ロケーションを登録する (コンソール)

> **Important**
>
> 次の手順では、Amazon S3 の場所がデータカタログと同じ AWS アカウントにあり、その場所のデータが暗号化されていないことを前提としています。クロスアカウント登録と暗号化されたロケーションの登録については、本章の他のセクションで説明されています。

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者、または `lakeformation:RegisterResource` IAM 許可を持つユーザーとしてサインインします。
2. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。
3. [Register location] (ロケーションを登録) を選択してから、[Browse](参照) を選択して Amazon Simple Storage Service (Amazon S3) パスを選択します。

4. (強く推奨されるオプション) [ロケーションのアクセス許可のレビュー] を選択して、選択した Amazon S3 ロケーションにあるすべての既存のリソースおよびアクセス許可のリストを確認します。

   選択されたロケーションの登録により、Lake Formation ユーザーがそのロケーションにすでに存在するデータにアクセスできるようになる可能性があります。このリストの確認は、既存のデータのセキュリティが確保されていることを確実にするために役立ちます。

5. [IAM role] (IAMロール) には、`AWSServiceRoleForLakeFormationDataAccess` サービスリンクロール (デフォルト)、または「the section called “ロケーションの登録に使用されるロールの要件”」の要件を満たすカスタム IAM ロールを選択します。

   登録したロケーションやその他の詳細を更新できるのは、カスタム IAM ロールを使用して登録した場合のみです。サービスにリンクされたロールを使用して登録したロケーションを編集するには、ロケーションの登録を解除して再度登録する必要があります。

6. データカタログフェデレーションを有効にするオプションを選択して、Lake Formation がロールを引き受け、統合 AWS サービスに一時的な認証情報を提供してフェデレーションデータベースのテーブルにアクセスできるようにします。ロケーションが Lake Formation に登録されていて、フェデレーションデータベースのテーブルにも同じロケーションを使用する場合は、同じロケーションを [Data Catalog フェデレーションを有効にする] オプションで登録する必要があります。

7. Lake Formation 許可をデフォルトで有効にしない場合は、[ハイブリッドアクセスモード] を選択します。Amazon S3 ロケーションをハイブリッドアクセスモードで登録すると、そのロケーションにあるデータベースとテーブルのプリンシパルをオプトインすることで、Lake Formation 許可を有効にできます。

   ハイブリッドアクセスモードアクセスの設定の詳細については、「ハイブリッドアクセスモード」を参照してください。

8. 所有者アカウントで S3 パスを登録を選択し、S3 バケット所有者アカウントを指定して、Lake Formation が Amazon S3 ロケーションで登録ロールのアクセスを確認し、Lake Formation が登録されたロケーションで Amazon S3 ファイルの一時的な認証情報を配信できるようにします。

9. [ロケーションを登録] を選択します。

ロケーションを登録するには (AWS CLI)

1. 新しいロケーションを Lake Formation に登録します。

   この例では、サービスにリンクされたロールを使用してロケーションを登録します。その代わりに `--role-arn` 引数を使用して、独自のロールを提供することができます。

   <s3-path> を有効な Amazon S3 パスに、アカウント番号を有効な AWS アカウントに置き換え、*<s3-access-role>* をデータロケーションを登録する権限を持つ IAM ロールに置き換えます。

   > **Note**
   >
   > ロケーションの登録にサービスにリンクされたロールを使用した場合、登録したロケーションのプロパティは編集できません。

   ```
   aws lakeformation register-resource \
    --resource-arn arn:aws:s3:::<s3-path> \
    --use-service-linked-role
   ```

   次の例では、カスタムロールを使用してロケーションを登録します。

   ```
   aws lakeformation register-resource \
    --resource-arn arn:aws:s3:::<s3-path> \
    --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role>
   ```

2. Lake Formation に登録したロケーションを更新するには

   登録したロケーションは、カスタム IAM ロールを使用して登録している場合にのみ編集できます。サービスにリンクされたロールに登録されているロケーションについては、ロケーションの登録を解除してから再度登録する必要があります。詳細については、「the section called “Amazon S3 ロケーションの登録解除”」を参照してください。

   ```
   aws lakeformation update-resource \
    --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role>\
    --resource-arn arn:aws:s3:::<s3-path>
   ```

```
aws lakeformation update-resource \
 --resource-arn arn:aws:s3:::<s3-path> \
 --use-service-linked-role
```

3. ハイブリッドアクセスモードでデータロケーションをフェデレーションに登録します。

```
aws lakeformation register-resource \
 --resource-arn arn:aws:s3:::<s3-path> \
 --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role> \
 --hybrid-access-enabled
```

```
aws lakeformation register-resource \
 --resource-arn arn:aws:s3:::<s3-path> \
 --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role> \
 --with-federation
```

```
aws lakeformation update-resource \
 --resource-arn arn:aws:s3:::<s3-path> \
 --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role> \
 --hybrid-access-enabled
```

4. 予想される所有者アカウントにロケーションを登録する

`--expected-resource-owner-account` フラグを使用して、Lake Formation が Amazon S3 ロケーションへの登録ロールのアクセスを検証し、登録されたロケーションで Amazon S3 ファイルの一時的な認証情報を提供できるようにします。

```
aws lakeformation register-resource \
 --resource-arn arn:aws:s3:::<s3-path> \
 --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role> \
 --expected-resource-owner-account <123456789012>
```

```
aws lakeformation update-resource \
 --resource-arn arn:aws:s3:::<s3-path> \
 --role-arn arn:aws:iam::<123456789012>:role/<s3-access-role> \
 --expected-resource-owner-account <123456789012>
```

詳細については、「RegisterResource」を参照してください。

> **Note**
>
> Amazon S3 の場所を登録すると、その場所 (またはその子の場所) を指す AWS Glue テーブルは、`GetTable`呼び出し`true`のように `IsRegisteredWithLakeFormation`パラメータの値を返します。`GetTables` および `SearchTables` などの Data Catalog API 操作が `IsRegisteredWithLakeFormation` パラメータの値を更新せず、デフォルト値の false を返すという既知の制限があります。`IsRegisteredWithLakeFormation` パラメータの正しい値を表示するには、`GetTable` API を使用することが推奨されます。

暗号化された Amazon S3 ロケーションの登録

Lake Formation は AWS Key Management Service (AWS KMS) と統合して、Amazon Simple Storage Service (Amazon S3) ロケーションにあるデータの暗号化と復号化を行うために、他の統合サービスをより簡単にセットアップできるようにします。

カスタマー管理 AWS KMS keys と の両方 AWS マネージドキー がサポートされています。現在、クライアント側の暗号化/復号は Athena でのみサポートされています。

Amazon S3 ロケーションを登録するときは、 AWS Identity and Access Management (IAM) ロールを指定する必要があります。 Amazon S3 暗号化された Amazon S3 の場所の場合、ロールには を使用してデータを暗号化および復号するアクセス許可が必要です。または AWS KMS key、KMS キーポリシーはキーに対するアクセス許可をロールに付与する必要があります。

> **Important**
>
> [Requester pays] (リクエスタ支払い) が有効になっている Amazon S3 バケットの登録は避けてください。Lake Formation に登録されたバケットの場合、バケットの登録に使用されるロールは常にリクエスト元であると見なされます。バケットが別の AWS アカウントからアクセスされた場合、ロールがバケット所有者と同じアカウントに属している場合、バケット所有者はデータアクセスに対して課金されます。

Lake Formation は、サービスにリンクされたロールを使用してデータの場所を登録します。ただし、このロールにはいくつかの制約があります。こうした制約があるため、柔軟性と制御性を高めるために、代わりにカスタム IAM ロールを作成して使用することをお勧めします。場所を登録するた

めに作成するカスタムロールは、the section called “ロケーションの登録に使用されるロールの要件” で指定された要件を満たしている必要があります。

> **Important**
>
> を使用して Amazon S3 の場所を AWS マネージドキー 暗号化した場合、Lake Formation サービスにリンクされたロールを使用することはできません。カスタムロールを使用して、キーに対する IAM 許可をロールに追加する必要があります。詳細については、このセクションで後ほど説明します。

以下の手順では、カスタマーマネージドキー、または AWS マネージドキーで暗号化された Amazon S3 ロケーションを登録する方法を説明します。

- カスタマーマネージドキーで暗号化されたロケーションの登録
- で暗号化された場所の登録 AWS マネージドキー

開始する前に

「ロケーションの登録に使用されるロールの要件」を確認してください。

カスタマーマネージドキーで暗号化された Amazon S3 ロケーションを登録する

> **Note**
>
> KMS キーまたは Amazon S3 の場所がデータカタログと同じ AWS アカウントにない場合は、the section called “AWS アカウント間で暗号化された Amazon S3 の場所を登録する”代わりに「」の手順に従います。

1. https://console.aws.amazon.com/kms で AWS KMS コンソールを開き、 AWS Identity and Access Management (IAM) 管理ユーザーとして、または場所の暗号化に使用される KMS キーのキーポリシーを変更できるユーザーとしてログインします。
2. ナビゲーションペインで [カスタマーマネージドキー] を選択してから、目的の KMS キーの名前を選択します。
3. KMS キーの詳細ページで [キーポリシー] タブを選択してから、以下のいずれかを行って、カスタムロールまたは Lake Formation サービスリンクロールを KMS キーユーザーとして追加します。

- デフォルトビューが表示されている場合 (キー管理者、キー削除、キーユーザー、その他の AWS アカウントセクションを含む) – キーユーザーセクションで、カスタムロール または Lake Formation サービスにリンクされたロール を追加しますAWSServiceRoleForLakeFormationDataAccess。
- キーポリシー (JSON) が表示されている場合 – 以下の例にあるように、ポリシーを編集して、「Allow use of the key」オブジェクトにカスタムロールまたは Lake Formation サービスリンクロール (AWSServiceRoleForLakeFormationDataAccess) を追加します。

> **Note**
>
> そのオブジェクトが欠落している場合は、例にある許可と共に追加してください。この例は、サービスリンクロールを使用しています。

```
        ...
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/aws-service-role/
lakeformation.amazonaws.com/AWSServiceRoleForLakeFormationDataAccess",
                    "arn:aws:iam::111122223333:user/keyuser"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        ...
```

4. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者、または `lakeformation:RegisterResource` IAM 許可を持つユーザーとしてサインインします。
5. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。
6. [Register location] (ロケーションを登録) を選択してから、[Browse](参照) を選択して Amazon Simple Storage Service (Amazon S3) パスを選択します。
7. (強く推奨されるオプション) [Review location permissions] (ロケーションの許可のレビュー) を選択して、選択された Amazon S3 ロケーションにあるすべての既存のリソースとそれらの許可のリストを確認します。

   選択されたロケーションの登録により、Lake Formation ユーザーがそのロケーションにすでに存在するデータにアクセスできるようになる可能性があります。このリストの確認は、既存のデータのセキュリティが確保されていることを確実にするために役立ちます。
8. [IAM role] (IAMロール) には、`AWSServiceRoleForLakeFormationDataAccess` サービスリンクロール (デフォルト)、または「the section called “ロケーションの登録に使用されるロールの要件”」に適合するカスタム IAM ロールを選択します。
9. [Register location] (ロケーションを登録) を選択します。

サービスリンクロールの詳細については、「Lake Formation のサービスリンクロールの許可」を参照してください。

で暗号化された Amazon S3 の場所を登録するには AWS マネージドキー

> **Important**
>
> Amazon S3 の場所がデータカタログと同じ AWS アカウントにない場合は、the section called “AWS アカウント間で暗号化された Amazon S3 の場所を登録する”代わりに「」の手順に従います。

1. ロケーションの登録に使用する IAM ロールを作成します。ロールが「the section called “ロケーションの登録に使用されるロールの要件”」に記載されている条件を満たすことを確認してください。
2. 以下のインラインポリシーをロールに追加します。これは、キーに対する許可をロールに付与します。`Resource` の仕様は、 AWS マネージドキーの Amazon リソースネーム (ARN) を指定する必要があります。ARN は AWS KMS コンソールから取得できます。正しい ARN を取得する

には、その場所の暗号化に使用された AWS マネージドキー と同じ AWS アカウントとリージョンで AWS KMS コンソールにログインしてください。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

キー ID の代わりに KMS キーエイリアスを使用できます - `arn:aws:kms:region:account-id:key/alias/your-key-alias`

詳細については、「 AWS Key Management Service デベロッパーガイド」の「 セクションのエイリア AWS KMSス」を参照してください。

3. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者、または `lakeformation:RegisterResource` IAM 許可を持つユーザーとしてサインインします。
4. ナビゲーションペインの [管理] で、[データレイクの口ケーション] を選択します。
5. [Register location] (口ケーションを登録) を選択してから、[Browse](参照) を選択して Amazon S3 パスを選択します。
6. (強く推奨されるオプション) [Review location permissions] (口ケーションの許可のレビュー) を選択して、選択された Amazon S3 口ケーションにあるすべての既存のリソースとそれらの許可のリストを確認します。

選択されたロケーションの登録により、Lake Formation ユーザーがそのロケーションにすでに存在するデータにアクセスできるようになる可能性があります。このリストの確認は、既存のデータのセキュリティが確保されていることを確実にするために役立ちます。

7. [IAM role] (IAM ロール) には、ステップ 1 で作成したロールを選択します。
8. [Register location] (ロケーションを登録) を選択します。

別の AWS アカウントでの Amazon S3 ロケーションの登録

AWS Lake Formation では、 AWS アカウント間で Amazon Simple Storage Service (Amazon S3) ロケーションを登録できます。たとえば、 AWS Glue Data Catalog がアカウント A にある場合、アカウント A のユーザーはアカウント B に Amazon S3 バケットを登録できます。

AWS アカウント A の AWS Identity and Access Management (IAM) ロールを使用してアカウント B に AWS Amazon S3 バケットを登録するには、次のアクセス許可が必要です。

- アカウント A のロールが、アカウント B のバケットに対する許可を付与する必要があります。
- アカウント B のバケットポリシーが、アカウント A のロールにアクセス許可を付与する必要があります。

> **Important**
>
> [Requester pays] (リクエスタ支払い) が有効になっている Amazon S3 バケットの登録は避けてください。Lake Formation に登録されたバケットの場合、バケットの登録に使用されるロールは常にリクエスト元であると見なされます。バケットが別の AWS アカウントからアクセスされた場合、ロールがバケット所有者と同じアカウントに属している場合、バケット所有者はデータアクセスに対して課金されます。
> Lake Formation サービスリンクロールを使用して、別のアカウントにあるロケーションを登録することはできません。その代わりに、ユーザー定義のロールを使用する必要があります。このロールは、「the section called “ロケーションの登録に使用されるロールの要件”」の要件を満たす必要があります。サービスリンクロールの詳細については、「Lake Formation のサービスリンクロールの許可」を参照してください。

[開始する前に]

「ロケーションの登録に使用されるロールの要件」を確認してください。

別の AWS アカウントでロケーションを登録するには

> **Note**
>
> ロケーションが暗号化されている場合は、代わりに「the section called “AWS アカウント間で暗号化された Amazon S3 の場所を登録する”」の手順を実行してください。

以下の手順は、Data Catalog が含まれるアカウント 1111-2222-3333 のプリンシパルが、アカウント 1234-5678-9012 にある Amazon S3 バケット `awsexamplebucket1` を登録したいという状況を前提としています。

1. アカウント 1111-2222-3333 で、 にサインイン AWS マネジメントコンソール し、 で IAM コンソールを開きますhttps://console.aws.amazon.com/iam/。
2. 新しいロールを作成するか、「the section called “ロケーションの登録に使用されるロールの要件”」の要件を満たす既存のロールを表示します。ロールが `awsexamplebucket1` に対する Amazon S3 許可を付与することを確認します。
3. Amazon S3 コンソール (https://console.aws.amazon.com/s3/) を開きます。アカウント 1234-5678-9012 でサインインします。
4. [Bucket name] (バケット名) リストで、`awsexamplebucket1` というバケット名を選択します。
5. [Permissions] (アクセス許可) を選択します。
6. [Permissions] (アクセス許可) ページで、[Bucket Policy] (バケットポリシー) を選択します。
7. [Bucket policy editor] (バケットポリシーエディタ) に、以下のポリシーを貼り付けます。*<role-name>* をロールの名前に置き換えます。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Principal": {
                "AWS":"arn:aws:iam::111122223333:role/<role-name>"
            },
            "Action":"s3:ListBucket",
```

```
            "Resource":"arn:aws:s3:::awsexamplebucket1"
        },
        {
            "Effect":"Allow",
            "Principal": {
                "AWS":"arn:aws:iam::111122223333:role/<role-name>"
            },
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource":"arn:aws:s3:::awsexamplebucket1/*"
        }
    ]
}
```

8. [保存] を選択します。
9. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者として、またはロケーションを登録するために十分な許可を持つユーザーとして、アカウント 1111-2222-3333 にサインインします。
10. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。
11. [データレイクのロケーション] ページで、[ロケーションを登録] を選択します。
12. [Register location] (ロケーションの登録) ページで、[Amazon S3 path] (Amazon S3 パス) にバケット名 `s3://awsexamplebucket1` を入力します。

> **Note**
>
> クロスアカウントバケットは [Browse] (参照) を選択してもリストに表示されないため、バケット名を入力する必要があります。

13. [IAM role] (IAM ロール) でロールを選択します。
14. [Register location] (ロケーションを登録) を選択します。

AWS アカウント間で暗号化された Amazon S3 の場所を登録する

AWS Lake Formation は AWS Key Management Service (AWS KMS) と統合されているため、Amazon Simple Storage Service (Amazon S3) ロケーションでデータを暗号化および復号するための他の統合サービスをより簡単にセットアップできます。

カスタマーマネージドキーと の両方 AWS マネージドキー がサポートされています。クライアント側の暗号化/復号化はサポートされていません。

> **Important**
>
> [Requester pays] (リクエスタ支払い) が有効になっている Amazon S3 バケットの登録は避けてください。Lake Formation に登録されたバケットの場合、バケットの登録に使用されるロールは常にリクエスト元であると見なされます。バケットが別の AWS アカウントからアクセスされた場合、ロールがバケット所有者と同じアカウントに属している場合、バケット所有者はデータアクセスに対して課金されます。

このセクションでは、以下の状況で Amazon S3 ロケーションを登録する方法について説明します。

- Amazon S3 内ロケーション内のデータが、 AWS KMSで作成された KMS キーで暗号化されている。
- Amazon S3 の場所が と同じ AWS アカウント内にありません AWS Glue Data Catalog。
- KMS キーは、 データカタログと同じ AWS アカウントにあるか、または存在しないかのいずれかです。

AWS アカウント A の (IAM) ロールを使用して AWS アカウント B に AWS KMS暗号化された Amazon S3 バケットを登録するには、 AWS Identity and Access Management 次のアクセス許可が必要です。

- アカウント A のロールが、アカウント B のバケットに対する許可を付与する必要があります。
- アカウント B のバケットポリシーが、アカウント A のロールにアクセス許可を付与する必要があります。
- KMS キーがアカウント B にある場合は、キーポリシーがアカウント A のロールにアクセス権を付与し、アカウント A のロールが KMS キーに対する許可を付与する必要があります。

次の手順では、データカタログを含む AWS アカウントにロールを作成します (前の説明のアカウント A)。次に、このロールを使用してロケーションを登録します。Lake Formation は、Amazon S3 内の基盤となるデータにアクセスするときに、このロールを引き受けます。引き受けたロールには、KMS キーに対する必要な許可があります。その結果、ETL ジョブや Amazon Athenaなどの統合サービスで基盤となるデータにアクセスするプリンシパルに、KMS キーに対する許可を付与する必要がなくなります。

> **Important**
>
> Lake Formation サービスリンクロールを使用して、別のアカウントにあるロケーションを登録することはできません。その代わりに、ユーザー定義のロールを使用する必要があります。このロールは、「the section called “ロケーションの登録に使用されるロールの要件”」の要件を満たす必要があります。サービスリンクロールの詳細については、「Lake Formation のサービスリンクロールの許可」を参照してください。

開始する前に

「ロケーションの登録に使用されるロールの要件」を確認してください。

AWS アカウント間で暗号化された Amazon S3 の場所を登録するには

1. データカタログと同じ AWS アカウントで、 にサインイン AWS マネジメントコンソール し、で IAM コンソールを開きますhttps://console.aws.amazon.com/iam/。
2. 新しいロールを作成するか、「the section called “ロケーションの登録に使用されるロールの要件”」の要件を満たす既存のロールを表示します。そのロールに、ロケーションに対する Amazon S3 許可を付与するポリシーが含まれていることを確認します。
3. KMS キーが Data Catalog と同じアカウントにないという場合は、KMS キーに対する必要な許可を付与するインラインポリシーをロールに追加します。以下は、ポリシーの例です。リージョンとアカウント ID を、KMS キーのリージョンとアカウント番号に置き換えます。*<key-id>* は、キー ID に置き換えます。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
         ],
```

```
        "Resource": "arn:aws:kms:us-east-1:111122223333:key/<key-id>"
        }
    ]
}
```

4. Amazon S3 コンソールで、必要な Amazon S3 の許可をロールに付与するバケットポリシーを追加します。以下は、バケットポリシーの例です。アカウント ID をデータカタログの AWS アカウント番号に、*<role-name>* をロール名に、*<bucket-name>* をバケットの名前に置き換えてください。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Principal": {
                "AWS":"arn:aws:iam::111122223333:role/<role-name>"
            },
            "Action":"s3:ListBucket",
            "Resource":"arn:aws:s3:::<bucket-name>"
        },
        {
            "Effect":"Allow",
            "Principal": {
                "AWS":"arn:aws:iam::111122223333:role/<role-name>"
            },
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource":"arn:aws:s3:::<bucket-name>/*"
        }
    ]
}
```

5. で AWS KMS、KMS キーのユーザーとしてロールを追加します。

a. https://console.aws.amazon.com/kms で AWS KMS コンソールを開きます。次に、管理者ユーザーとして、またはロケーションの暗号化に使用された KMS キーのキーポリシーを変更できるユーザーとしてサインインします。

b. ナビゲーションペインで [Customer managed keys] (カスタマー管理型のキー) を選択してから、KMS キーの名前を選択します。

c. KMS キーの詳細ページの [Key policy] (キーポリシー) タブにキーポリシーの JSON ビューが表示されていない場合は、[Switch to policy view] (ポリシービューへの切り替え) を選択します。

d. [Key policy] (キーポリシー) セクションで [Edit] (編集) を選択し、以下の例にあるように、ロールの Amazon リソースネーム (ARN) を `Allow use of the key` オブジェクトに追加します。

> **Note**
>
> そのオブジェクトが欠落している場合は、例にある許可と共に追加してください。

```
...
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::<catalog-account-id>:role/<role-name>"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
...
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

6. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者として Data Catalog AWS アカウントにサインインします。
7. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。
8. [Register location] (ロケーションを登録) を選択します。
9. [Register location] (ロケーションの登録) ページの [Amazon S3 path] (Amazon S3 パス) に、ロケーションのパスを **s3://*<bucket>*/*<prefix>*** として入力します。*<bucket>* はバケット名、*<prefix>* はロケーションのパスの残りの部分に置き換えてください。

> **Note**
>
> クロスアカウントバケットは [Browse] (参照) を選択してもリストに表示されないため、パスを入力する必要があります。

10. [IAM role] (IAMロール) には、ステップ 2 からのロールを選択します。
11. [Register location] (ロケーションを登録) を選択します。

Amazon S3 ロケーションの登録解除

Amazon Simple Storage Service (Amazon S3) ロケーションを Lake Formation で管理する必要がなくなった場合は、このロケーションの登録を解除できます。ロケーションの登録を解除しても、そのロケーションに対して付与されている Lake Formation データロケーション許可には影響しません。登録を解除したロケーションは再登録でき、データロケーション許可は引き続き有効になります。ロケーションは、別のロールを使用して再登録できます。

ロケーションの登録を解除する (コンソール)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者、または `lakeformation:RegisterResource` IAM 許可を持つユーザーとしてサインインします。
2. ナビゲーションペインの [管理] で、[データレイクのロケーション] を選択します。
3. ロケーションを選択し、[Actions] (アクション) メニューで [Remove] (削除) を選択します。
4. 確認を求めるプロンプトが表示されたら、[Remove] (削除) を選択します。

Amazon S3 ロケーションへのアクセス

AWS Lake Formation では、 AWS Glue Data Catalog (データカタログ) に登録されたテーブルの Amazon Simple Storage Service (Amazon S3) の基盤となるデータファイルを読み書きできます。これにより、既存の Lake Formation テーブル許可を使用した SQL クエリと直接ファイルアクセスの両方に対する単一のアクセス許可セットが提供されます。Lake Formation は、認証情報供給機能を Data Catalog に登録されている Amazon S3 ロケーションに拡張します。アプリケーションが Data Catalog テーブルの基盤となる Amazon S3 ファイルへのアクセスをリクエストすると、Lake Formation は発信者の既存のテーブルレベルのアクセス許可を評価し、承認された場合は、そのテーブルの登録場所の有効期間が短いスコープの Amazon S3 認証情報を返します。

Amazon S3 ロケーションへの直接アクセスを有効にする

この機能を有効にするには、次のステップを完了する必要があります。

前提条件

機能フラグを有効にする前に、次の前提条件を完了してください。

- テーブルの場所を指定する – テーブル属性の LOCATION プロパティを使用して、Amazon S3 のデータへのパスを指定します。場所形式の詳細については、Amazon S3のテーブルの場所」を参照してください。 Amazon Athena
- Amazon S3 ロケーションの登録 – Amazon S3 ロケーションのデータカタログにテーブルをカタログ化し、Amazon S3 バケット所有者アカウントを指定して Lake Formation に登録します。ロケーションを登録すると、その Amazon S3 パスと、そのパスの下にあるすべてのフォルダが登録されます。詳細については、「Amazon S3 ロケーションの登録」を参照してください。
- Lake Formation 許可の付与 – Lake Formation コンソールまたは API を使用して、データサイエンティストとアプリケーションにテーブルに対する Lake Formation SELECT (または SUPER) 許可を付与します。 APIs Athena や Amazon EMR などのサービスでのテーブルアクセスに Lake Formation を既に使用している場合、これは既に設定されています。
- フルテーブルアクセスによるアプリケーション統合を有効にする – フルテーブルアクセスによるアプリケーション統合を有効にして、登録されたテーブルの場所への認証情報供給を許可します。詳細については、「フルテーブルアクセスのためのアプリケーション統合」を参照してください。

機能フラグを有効にする

サポートされているエンジンで `fs.s3a.lakeformation.access.grants.enabled`フラグを有効にします。

```
fs.s3a.lakeformation.access.grants.enabled = true
```

- Amazon EC2 での Amazon EMR – EMR 7.13 以降
- Amazon EMR on EKS – EMR 7.13 以降
- Amazon EMR Serverless – EMR 7.13 以降

設定が完了すると、標準 APIs を使用して EMR からデータファイルをすぐに読み書きできます。

Amazon EMR 統合の詳細については、「Amazon EMR 管理ガイド」の「Lake Formation パスベースの認証情報供給」を参照してください。

仕組み

Lake Formation ベースの Amazon S3 ロケーションアクセスは、次のフローに従います。

1. プリンシパルまたはロールは、Amazon EMR Spark ジョブやデータ処理パイプラインなどのサービスを通じて Amazon S3 データファイルへのアクセスをリクエストします。
2. EMR と統合された AWS SDK プラグインは、Amazon S3 リクエストをインターセプトし、Lake Formation GetTemporaryDataLocationCredentials API を呼び出します。
3. Lake Formation は、Amazon S3 パスが登録済みロケーションとカタログ化されたテーブルに対応しているかどうかをチェックし、関連するAWS Glueテーブルに対する発信者のアクセス許可を評価します。
4. 呼び出し元がフルテーブルアクセスを持つ対応するテーブルに対して SELECTまたは SUPER アクセス許可を持っている場合、Lake Formation はその Amazon S3 ロケーションを対象とする一時的な認証情報を提供します。 は READ 認証情報SELECTを付与し、 は READWRITE 認証情報SUPERを付与します。
5. プラグインはこれらの認証情報を使用して Amazon S3 リクエストを完了し、Amazon S3 はデータをアプリケーションに提供します。
6. 認証情報供給オペレーションはすべて CloudTrail に記録され、データアクセスの監査証跡を提供します。

このプロセスは透過的であり、標準のファイルベースの APIs で既存の Lake Formation アクセス許可を使用するだけです。

```
# Read raw data (Lake Formation-based S3 Location access)
```

```
raw_df = spark.read.json("s3://finance-datalake/raw/transactions/dt=2024-03-21/")

# Read governed data (Lake Formation-based S3 Location access)
transactions_df = spark.read.parquet("s3://data-lake/transactions/year=2026/")

# Write processed data (Lake Formation-based S3 Location access)
processed_df.write \
    .mode("append") \
    .partitionBy("transaction_date") \
    .parquet("s3://finance-datalake/processed/transactions/")

print("ETL complete. Records written:", processed_df.count())
```

アクセス許可の要件

この機能は現在、呼び出し元が完全なテーブルアクセス権を持っている場合にのみ認証情報を提供します。つまり、行または列フィルターのないすべての列SELECTに対して認証情報を提供します。テーブルに行レベルまたは列レベルのフィルターが適用されている場合、発信者はそれらのフィルターを適用できる Athena、Amazon EMR、AWS Glue、Amazon Redshift などの信頼できるエンジンを引き続き使用する必要があります。これにより、セキュリティ境界が一貫して維持されます。

クロスアカウントアクセス

Lake Formation はAmazon S3ロケーションへのアクセスを簡素化し、クロスアカウント共有機能を使用します。Lake Formation を介して別の AWS アカウントとテーブルを共有する場合、受信者は、同じ完全なテーブルアクセス要件に従って、Lake Formation アクセス許可を使用して基盤となる Amazon S3 データファイルにアクセスできます。

この機能は、クロスアカウントアクセスのための Lake Formation のリソースリンクメカニズムをサポートしています。コンシューマーアカウントに共有テーブルに対するアクセス許可がある場合、Lake Formation は登録済みアカウントの Amazon S3 ロケーションに限定された認証情報を提供し、個別の Amazon S3 バケットポリシーやクロスアカウント IAM ロールを必要とせずに、シームレスなクロスアカウントデータアクセスを可能にします。

ネストされた Amazon S3 の場所

複数のテーブルが同じバケット内のネストされた場所を指す場合、Lake Formation は次の動作を適用します。

- にアクセスするとs3://bucket、バケットレベルで登録されたテーブルに対応するアクセス許可が付与されます。

- にアクセスするとs3://bucket/folder1、その特定のパスに登録されているテーブルに対応するアクセス許可が付与されます。
- 登録されたテーブルがないフォルダ (などs3://bucket/folder2) にアクセスするには、最も近い親登録場所からアクセス許可を受け取ります。
- 同じ場所に複数のテーブルが登録されている場合、Lake Formation はアクセス許可の競合によりエラーを返します。

監査とコンプライアンス

認証情報供給オペレーションはすべて CloudTrail に記録され、データアクセスの監査証跡を提供します。Lake Formation が `GetTemporaryDataLocationCredentials` API を介して認証情報を提供すると、CloudTrail は以下を記録します。

- プリンシパル (ユーザーまたはロール)
- タイムスタンプ
- Amazon S3 ロケーション
- 関連付けられたAWS Glueテーブル

これらの認証情報を使用して行われた後続の Amazon S3 API コールも Amazon S3 データイベントとして CloudTrail に記録され、コンテキストによって Lake Formation アクセス許可の付与にリンクされます。これにより、監査人は、どのユーザーがどのデータ、いつ、どのアクセス許可パスにアクセスしたかを完全に単一の CloudTrail ログストリームで把握できます。

サポートされている AWS サービス

Lake Formation はAmazon S3ロケーションへのアクセスを簡素化し、以下と連携します。

- Amazon EC2 での Amazon EMR (EMR 7.13 以降)
- Amazon EMR on EKS (EMR 7.13 以降)
- Amazon EMR Serverless (EMR 7.13 以降)

サードパーティーサービス用のオープンソースプラグイン

API または が提供するオープンソースプラグインを使用して APIs 、 Apache Spark または Trinoアプリケーションを統合することもできます AWS。詳細については、GitHub のaws-lakeformation-accessgrants-plugin-java-v2」を参照してください。

考慮事項

Amazon S3 ロケーションで認証情報供給を使用する場合は、次の考慮事項に注意してください。

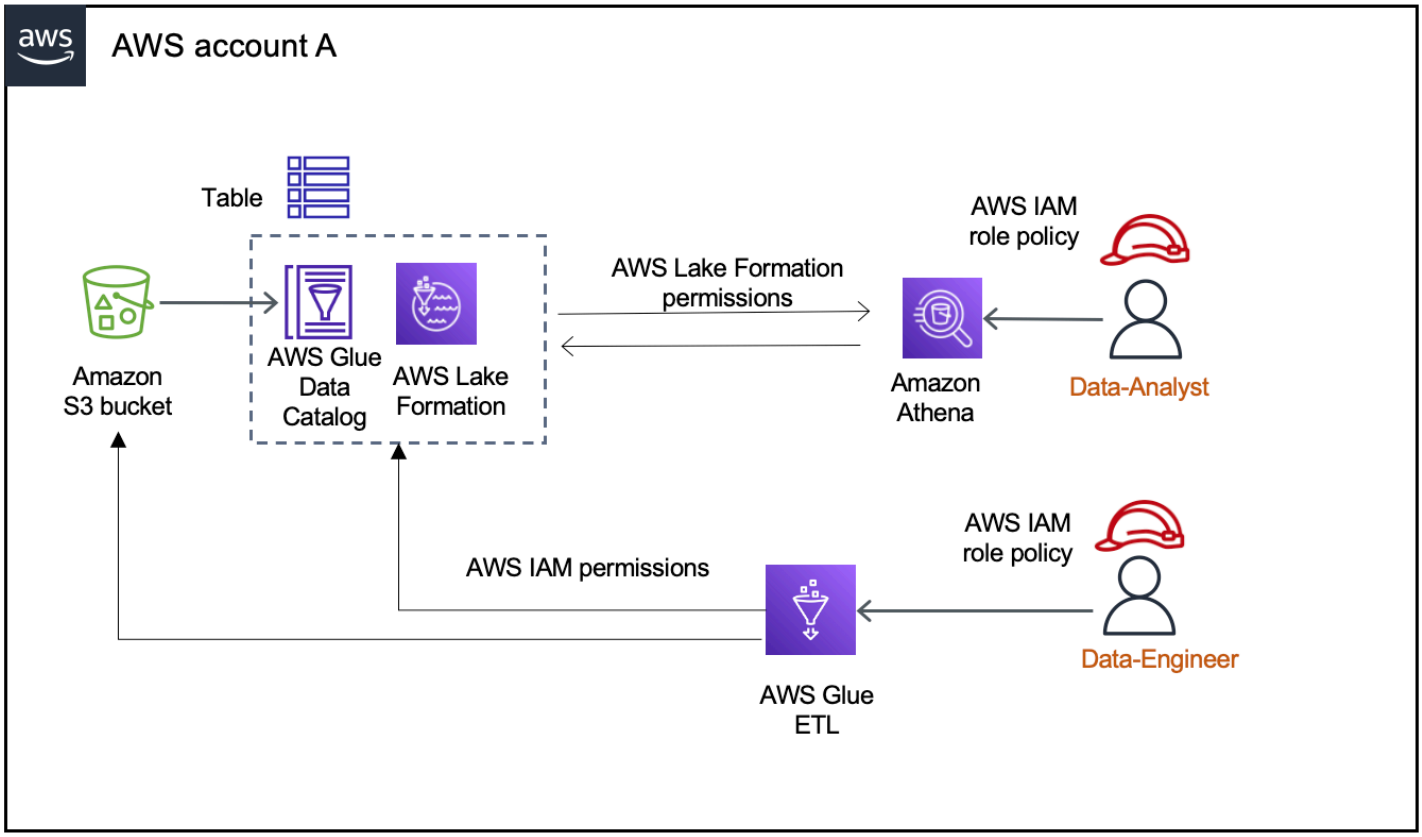
- Amazon S3 ロケーションの認証情報供給は、クロスリージョンではサポートされていません。
- 認証情報供給は、テーブルのプライマリロケーションとして含まれる Amazon S3 ロケーションでサポートされています。
- プラグインは、Apache Hive、Apache Hudi、および Delta Lakeテーブル形式をサポートしています。 Apache Icebergは現在サポートされていません。
- プラグインは現在、Amazon EMR Spark Fine-Grained Access Control (FGAC) モードではサポートされていません。

ハイブリッドアクセスモード

AWS Lake Formation ハイブリッドアクセスモードは、同じ AWS Glue Data Catalog オブジェクトへの 2 つのアクセス許可パスをサポートします。
最初のパスでは、Lake Formation を使用して特定のプリンシパルを選択し、オプトインしてカタログ、データベース、テーブル、ビューにアクセスするための Lake Formation アクセス許可を付与できます。2 番目のパスでは、他のすべてのプリンシパルが Amazon S3 および AWS Glue アクションのデフォルトの IAM プリンシパルポリシーを介してこれらのリソースにアクセスできます。

Amazon S3 ロケーションを Lake Formation に登録する場合、その口ケーションのすべてのリソースに Lake Formation 許可を適用するか、ハイブリッドアクセスモードを使用するかを選択できます。ハイブリッドアクセスモードは、デフォルトで、`CREATE_TABLE`、`CREATE_PARTITION`、および `UPDATE_TABLE` 許可のみが適用されます。Amazon S3 ロケーションがハイブリッドモードの場合、その口ケーションの Data Catalog オブジェクトのプリンシパルをオプトインすることで、Lake Formation アクセス許可を有効にできます。
これは、Lake Formation アクセス許可と IAM アクセス許可の両方が、そのデータへのアクセスを制御できることを意味します。つまり、オプトインされたプリンシパルはデータにアクセスするために Lake Formation アクセス許可と IAM アクセス許可の両方を必要とし、non-opted-inプリンシパルは引き続き IAM アクセス許可のみを使用してデータにアクセスします。

したがって、ハイブリッドアクセスモードでは、他の既存のユーザーやワークロードのアクセスを中断することなく、特定のユーザーのセットに対してデータカタログ内のカタログ、データベース、およびテーブルに対して Lake Formation を選択的に有効にする柔軟性が得られます。



考慮事項と制限事項については、「ハイブリッドアクセスモードには次の考慮事項と制限事項が適用されます。」を参照してください。

用語と定義

アクセス許可の設定方法に基づく Data Catalog リソースの定義は次のとおりです。

Lake Formation のリソース

Lake Formation に登録されているリソース。ユーザーがリソースにアクセスするには、Lake Formation 許可が必要です。

AWS Glue リソース

Lake Formation に登録されていないリソース。リソースに `IAMAllowedPrincipals` グループのアクセス許可があるため、リソースにアクセスするには IAM 許可のみが必要です。Lake Formation 許可は適用されません。

`IAMAllowedPrincipals` グループのアクセス許可の詳細については、「メタデータアクセス許可」を参照してください。

ハイブリッドリソース

ハイブリッドアクセスモードで登録されたリソース。リソースにアクセスするユーザーに基づいて、リソースは Lake Formation リソースと AWS Glue リソースの間で動的に切り替わります。

一般的なハイブリッドアクセスモードのユースケース

ハイブリッドアクセスモードを使用すると、単一アカウントおよびクロスアカウントのデータ共有シナリオでアクセスを許可できます。

単一アカウントのシナリオ

- AWS Glue リソースをハイブリッドリソースに変換する – このシナリオでは、現在 Lake Formation を使用していませんが、Data Catalog オブジェクトに Lake Formation アクセス許可を採用したいと考えています。Amazon S3 ロケーションをハイブリッドアクセスモードで登録すると、そのロケーションを指す特定のデータベースとテーブルをオプトインするユーザーに、Lake Formation 許可を付与できます。
- Lake Formation リソースをハイブリッドリソースに変換する - 現在、Lake Formation アクセス許可を使用してデータカタログデータベースへのアクセスを制御していますが、既存の Lake Formation アクセス許可を中断せずに、Amazon S3 と AWS Glue の IAM アクセス許可を使用して新しいプリンシパルにアクセスを許可したいと考えています。

  データロケーション登録をハイブリッドアクセスモードに更新すると、新しいプリンシパルは、既存のユーザーの Lake Formation 許可を中断することなく、IAM 許可ポリシーを使用して Amazon S3 ロケーションを指す Data Catalog データベースにアクセスできます。

  データロケーション登録を更新してハイブリッドアクセスモードを有効にする前に、まず、現在 Lake Formation 許可でリソースにアクセスしているプリンシパルをオプトインする必要があります。
  これは、現在のワークフローが中断される可能性を防ぐためです。
  また、データベース内のテーブルに対する `Super` 許可を `IAMAllowedPrincipal` グループに付与する必要があります。

クロスアカウントデータ共有のシナリオ

- ハイブリッドアクセスモードを使用して AWS Glue リソースを共有する – このシナリオでは、プロデューサーアカウントには、Amazon S3 および AWS Glue アクションの IAM アクセス許可ポリ

シーを使用して、コンシューマーアカウントと現在共有されているデータベースにテーブルがあります。データベースのデータロケーションは、Lake Formation に登録されていません。

ハイブリッドアクセスモードでデータロケーションを登録する前に、[クロスアカウントバージョン設定] をバージョン 4 に更新する必要があります。バージョン 4 では、IAMAllowedPrincipalグループにリソースに対する AWS RAM アクセス許可がある場合に、クロスアカウント共有に必要な新しいSuperアクセス許可ポリシーが提供されます。IAMAllowedPrincipal グループアクセス許可のあるリソースについては、外部アカウントに Lake Formation 許可を付与し、そのアカウントが Lake Formation 許可を使用するようにオプトインできます。受信者アカウントのデータレイク管理者は、アカウント内のプリンシパルに Lake Formation 許可を付与し、プリンシパルをオプトインして Lake Formation 許可を適用できます。

- ハイブリッドアクセスモードを使用して Lake Formation リソースを共有する – 現在、プロデューサーアカウントのデータベース内のテーブルは、Lake Formation 許可を適用するコンシューマーアカウントと共有されています。データベースのデータロケーションは、Lake Formation に登録されています。

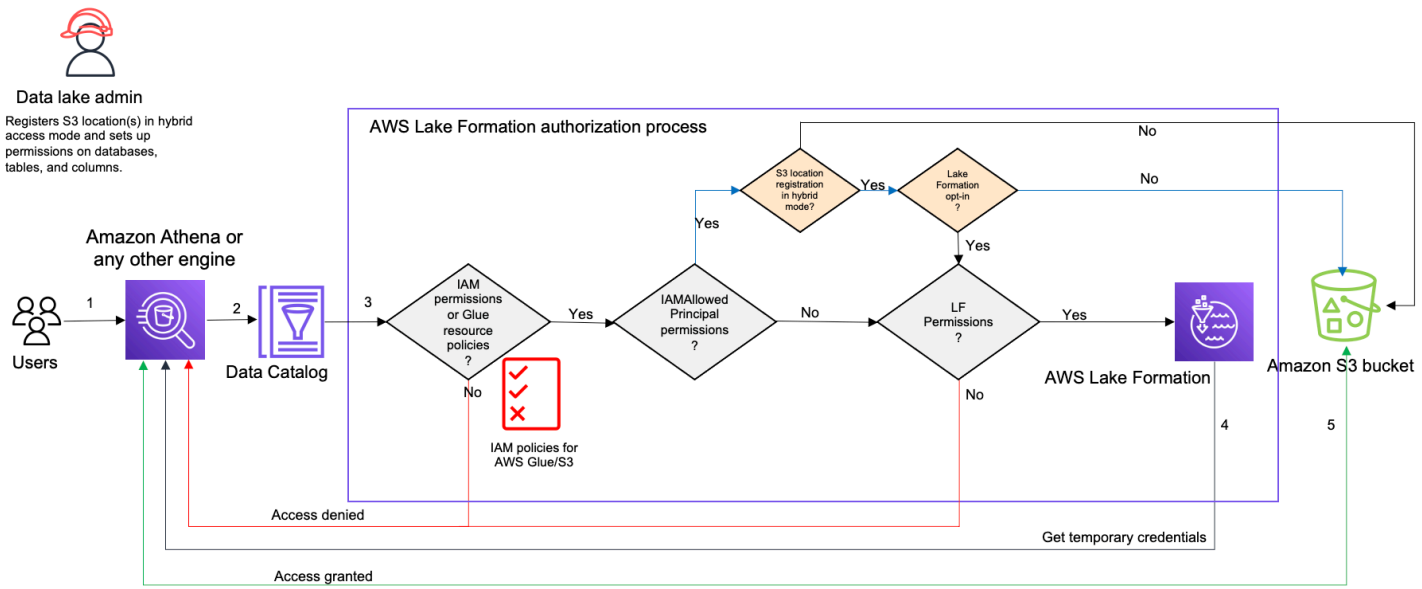
  この場合、Amazon S3 ロケーションの登録をハイブリッドアクセスモードに更新し、Amazon S3 バケットポリシーと Data Catalog リソースポリシーを使用して Amazon S3 のデータと Data Catalog のメタデータをコンシューマーアカウントのプリンシパルと共有できます。Amazon S3 ロケーションの登録を更新する前に、既存の Lake Formation 許可を再度付与し、プリンシパルをオプトインする必要があります。また、データベース内のテーブルに対する Super 許可を IAMAllowedPrincipals グループに付与する必要があります。

トピック

- ハイブリッドアクセスモードの仕組み
- ハイブリッドアクセスモードの設定 - 一般的なシナリオ
- ハイブリッドアクセスモードからプリンシパルとリソースを削除する
- ハイブリッドアクセスモードでプリンシパルとリソースを表示する
- その他のリソース

ハイブリッドアクセスモードの仕組み

次の図は、ハイブリッドアクセスモードで Data Catalog リソースにクエリを実行するときに Lake Formation 認可がどのように機能するかを示しています。



データレイク内のデータにアクセスする前に、データレイク管理者または管理権限を持つユーザーが、Data Catalog テーブルへのアクセスを許可または拒否する個々の Data Catalog テーブルのユーザーポリシーを設定します。次に、`RegisterResource` オペレーションを実行するアクセス許可を持つプリンシパルが、ハイブリッドアクセスモードで Lake Formation にテーブルの Amazon S3 ロケーションを登録します。データロケーションが Lake Formation に登録されていない場合、管理者は Data Catalog データベースとテーブルの特定のユーザーに Lake Formation アクセス許可を付与し、ハイブリッドアクセスモードでそれらのデータベースとテーブルに Lake Formation アクセス許可を使用するようにオプトインします。

1. クエリを送信する - プリンシパルは、Amazon Athena、Amazon EMR AWS Glue、Amazon Redshift Spectrum などの統合サービスを使用してクエリまたは ETL スクリプトを送信します。
2. データのリクエスト – 統合分析エンジンは、要求されているテーブルを識別し、メタデータのリクエストを Data Catalog (`GetTable`、`GetDatabase`) に送信します。
3. アクセス許可を確認 – Data Catalog は、クエリ元プリンシパルのアクセス許可を Lake Formation で検証します。
   a. テーブルに `IAMAllowedPrincipals` グループアクセス許可がアタッチされていない場合は、Lake Formation 許可が適用されます。
   b. プリンシパルがハイブリッドアクセスモードで Lake Formation 許可を使用することをオプトインしていて、テーブルに `IAMAllowedPrincipals` グループアクセス許可がアタッチされている場合、Lake Formation 許可が適用されます。クエリエンジンは、Lake Formation から受け取ったフィルターを適用し、データをユーザーに返します。

c. テーブルロケーションが Lake Formation に登録されておらず、プリンシパルがハイブリッドアクセスモードで Lake Formation 許可を使用することをオプトインしていない場合、Data Catalog はテーブルに `IAMAllowedPrincipals` グループアクセス許可がアタッチされているかどうかを確認します。このアクセス許可がテーブルに存在する場合、アカウント内のすべてのプリンシパルはテーブルに対する `Super` または `All` 許可が付与されます。

   データの場所が Lake Formation に登録されていない限り、オプトインしても Lake Formation の認証情報供給は利用できません。

4. 認証情報の取得 – Data Catalog は、テーブルのロケーションが Lake Formation に登録されているかどうかを確認し、エンジンに知らせます。基盤となるデータが Lake Formation に登録されている場合、分析エンジンは、Amazon S3 バケットのデータにアクセスするための一時的な認証情報を Lake Formation に要求します。
5. データの取得 – プリンシパルがテーブルデータへのアクセスを許可されている場合、Lake Formation は統合分析エンジンへの一時的なアクセスを提供します。一時的なアクセスを使用して、分析エンジンは Amazon S3 からデータを取得し、列、行、またはセルのフィルタリングなど、必要なフィルタリングを実行します。エンジンはジョブの実行を終了すると、結果をユーザーに返します。このプロセスは、認証情報の供給と呼ばれます。詳細については、「Lake Formation との統合」を参照してください。
6. テーブルのデータロケーションが Lake Formation に登録されていない場合、分析エンジンからの 2 回目の呼び出しは Amazon S3 に対して直接行われます。関係する Amazon S3 バケットポリシーと IAM ユーザーポリシーのデータアクセスが評価されます。IAM ポリシーを使用するときは、常に IAM のベストプラクティスに従うようにしてください。詳細については、「IAM ユーザーガイド」の「IAM でのセキュリティベストプラクティス」を参照してください。

ハイブリッドアクセスモードの設定 - 一般的なシナリオ

Lake Formation のアクセス許可と同様に、ハイブリッドアクセスモードを使用してデータアクセスを管理するシナリオには、通常、1 つの 内のプリンシパルへのアクセス AWS アカウント を提供し、外部 AWS アカウント またはプリンシパルへのアクセスを提供するという 2 種類があります。

このセクションでは、以下のシナリオでハイブリッドアクセスモードを設定する方法について説明します。

ハイブリッドアクセスモードで 1 つの 内でアクセス許可を管理する AWS アカウント

- AWS Glue リソースをハイブリッドリソースに変換する – 現在、Amazon S3 の IAM アクセス許可を使用してアカウント内のすべてのプリンシパルにデータベース内のテーブルへのアクセスを提供していますが、アクセス許可を段階的に管理するために Lake Formation を採用したいと考えています。 AWS Glue

- Lake Formation リソースをハイブリッドリソースに変換する - 現在、アカウント内のすべてのプリンシパルについて、Lake Formation を使用してデータベース内のテーブルへのアクセスを管理しているが、特定のプリンシパルにのみ Lake Formation を使用したいと考えている。同じデータベースとテーブルで AWS Glue と Amazon S3 の IAM アクセス許可を使用して、新しいプリンシパルへのアクセスを提供する必要があります。

間のハイブリッドアクセスモードでのアクセス許可の管理 AWS アカウント

- ハイブリッドアクセスモードを使用した AWS Glue リソースの共有 - 現在、テーブルのアクセス許可管理に Lake Formation を使用していないが、Lake Formation アクセス許可を適用して別のアカウント内のプリンシパルにアクセスを許可したいと考えている。
- ハイブリッドアクセスモードを使用して Lake Formation リソースを共有する – Lake Formation を使用してテーブルのアクセスを管理しているが、同じデータベースとテーブルで AWS Glue と Amazon S3 の IAM アクセス許可を使用して、別のアカウントのプリンシパルにアクセスを許可したい。

ハイブリッドアクセスモードの設定 – 概要ステップ

1. [ハイブリッドアクセスモード] を選択して、Amazon S3 データロケーションを Lake Formation に登録します。
2. プリンシパルは、Data Catalog のテーブルまたはデータベースのポイント先となるデータレイクの ロケーションに対する `DATA_LOCATION` 許可を持っている必要があります。
3. [クロスアカウントバージョン設定] をバージョン 4 に設定します。
4. データベースやテーブル上の特定の IAM ユーザーまたはロールにきめ細かいアクセス許可を付与します。同時に、データベース上の `IAMAllowedPrincipals` グループとデータベース内のすべてまたは選択したテーブルに、必ず `Super` または `All` 許可を設定します。

5. プリンシパルとリソースをオプトインします。アカウントの他のプリンシパルは、 および Amazon S3 アクションの IAM アクセス許可ポリシーを使用して、データベース AWS Glue とテーブルに引き続きアクセスできます。
6. オプションで、Lake Formation 許可を使用するようオプトインしているプリンシパルの Amazon S3 の IAM 許可ポリシーをクリーンアップします。

ハイブリッドアクセスモードの設定の前提条件

ハイブリッドアクセスモードを設定するための前提条件は次のとおりです。

> **Note**
>
> Lake Formation 管理者が Amazon S3 ロケーションをハイブリッドアクセスモードで登録し、プリンシパルとリソースをオプトインすることをお勧めします。

1. データロケーション許可 (DATA_LOCATION_ACCESS) は、Amazon S3 ロケーションをポイントする Data Catalog リソースを作成する場合に付与します。データロケーションのアクセス許可は、特定の Amazon S3 ロケーションを指す Data Catalog カタログ、データベース、テーブルを作成する機能を制御します。
2. Data Catalog リソースをハイブリッドアクセスモードの別のアカウントと共有するには (リソースからIAMAllowedPrincipalsグループアクセス許可を削除せずに)、クロスアカウントバージョン設定をバージョン 4 以降に更新する必要があります。Lake Formation コンソールを使用してバージョンを更新するには、データカタログ設定ページのクロスアカウントバージョン設定でバージョン 4 またはバージョン 5 を選択します。

   put-data-lake-settings AWS CLI コマンドを使用して、 CROSS_ACCOUNT_VERSIONパラメータをバージョン 4 または 5 に設定することもできます。

```
aws lakeformation put-data-lake-settings --region us-east-1 --data-lake-settings
 file://settings
{
"DataLakeAdmins": [
        {
"DataLakePrincipalIdentifier": "arn:aws:iam::<111122223333>:user/<user-name>"
        }
    ],
    "CreateDatabaseDefaultPermissions": [],
    "CreateTableDefaultPermissions": [],
```

```
    "Parameters": {
"CROSS_ACCOUNT_VERSION": "5"
    }
}
```

3. ハイブリッドアクセスモードでクロスアカウントアクセス許可を付与するには、付与者に AWS Glue および AWS RAM サービスに必要な IAM アクセス許可が必要です。 AWS 管理ポリシーは、必要なアクセス許可AWSLakeFormationCrossAccountManagerを付与します。
ハイブリッドアクセスモードでクロスアカウントデータ共有を有効にするために、次の 2 つの新しい IAM アクセス許可を追加して AWSLakeFormationCrossAccountManager 管理ポリシーを更新しました。

   - ram:ListResourceSharePermissions
   - ram:AssociateResourceSharePermission

   > **Note**
   >
   > 付与者ロールに AWS 管理ポリシーを使用していない場合は、カスタムポリシーに上記のポリシーを追加します。

Amazon S3 バケットの場所とユーザーアクセス

でカタログ、データベース、またはテーブルを作成するときに AWS Glue Data Catalog、基盤となるデータの Amazon S3 バケットの場所を指定し、Lake Formation に登録できます。以下の表は、テーブルまたはデータベースの Amazon S3 データの場所に基づいて、 AWS Glue および Lake Formation ユーザー (プリンシパル) のアクセス許可がどのように機能するかを示しています。

Lake Formation に登録された Amazon S3 ロケーション

データベースの Amazon S3 ロケーション	AWS Glue ユーザー	Lake Formation ユーザー
Lake Formation に登録 (ハイブリッドアクセスモードまたは Lake Formation モード)	IAMAllowedPrincipals グループ (スーパーアクセス) のアクセス許可を継承し、Amazon S3 データロケーションへの読	付与された CREATE TABLE アクセス許可から、テーブルを作成するアクセス許可を継承します。

データベースの Amazon S3 ロケーション	AWS Glue ユーザー	Lake Formation ユーザー
	み取り/書き込みアクセス権を持ちます。	
関連付けられた Amazon S3 ロケーションなし	CREATE TABLE および INSERT TABLE ステートメントを実行するには、明示的な DATA LOCATION アクセス許可が必要です。	CREATE TABLE および INSERT TABLE ステートメントを実行するには、明示的な DATA LOCATION アクセス許可が必要です。

IsRegisteredWithLakeFormation テーブルプロパティ

テーブルの `IsRegisteredWithLakeFormation` プロパティは、テーブルのデータロケーションがリクエスタの Lake Formation に登録されているかどうかを示します。ロケーションのアクセス許可モードが Lake Formation として登録されている場合は、すべてのユーザーがそのテーブルにオプトインされていると見なされるため、データロケーションにアクセスするすべてのユーザーに対して `IsRegisteredWithLakeFormation` プロパティが `true` になります。ロケーションがハイブリッドアクセスモードで登録されている場合は、そのテーブルにオプトインしたユーザーに対してのみ値が `true` に設定されます。

`IsRegisteredWithLakeFormation` の仕組み

アクセス許可モード	ユーザー/ロール	**`IsRegisteredWithLakeFormation`**	説明
Lake Formation	すべて	正	ロケーションが Lake Formation で登録されている場合、`IsRegisteredWithLakeFormation` プロパティはすべてのユーザーに対して true に設定されます。つまり、Lake

アクセス許可モード	ユーザー/ロール	**IsRegisteredWithLakeFormation**	説明
			Formation で定義されたアクセス許可が、登録されたロケーションに適用されます。認証情報供給は Lake Formation によって行われます。
ハイブリッドアクセスモード	オプトイン済み	正	テーブルのデータアクセスとガバナンスに Lake Formation を使用するようにオプトインしたユーザーでは、そのテーブルの `IsRegisteredWithLakeFormation` プロパティが `true` に設定されます。これらのユーザーには、登録されているロケーションに対して Lake Formation で定義されたアクセス許可ポリシーが適用されます。

アクセス許可モード	ユーザー/ロール	**IsRegisteredWithLakeFormation**	説明
ハイブリッドアクセスモード	オプトインなし	誤	Lake Formation アクセス許可の使用にオプトインしていないユーザーの場合、IsRegisteredWithLakeFormation プロパティは false に設定されます。これらのユーザーには、登録されている場所に対して Lake Formation で定義されているアクセス許可ポリシーは適用されません。代わりに、ユーザーは Amazon S3 アクセス許可ポリシーに従います。

AWS Glue リソースをハイブリッドリソースに変換する

以下のステップに従って Amazon S3 ロケーションをハイブリッドアクセスモードで登録し、既存の Data Catalog ユーザーのデータアクセスを中断することなく、新しい Lake Formation ユーザーをオンボーディングします。

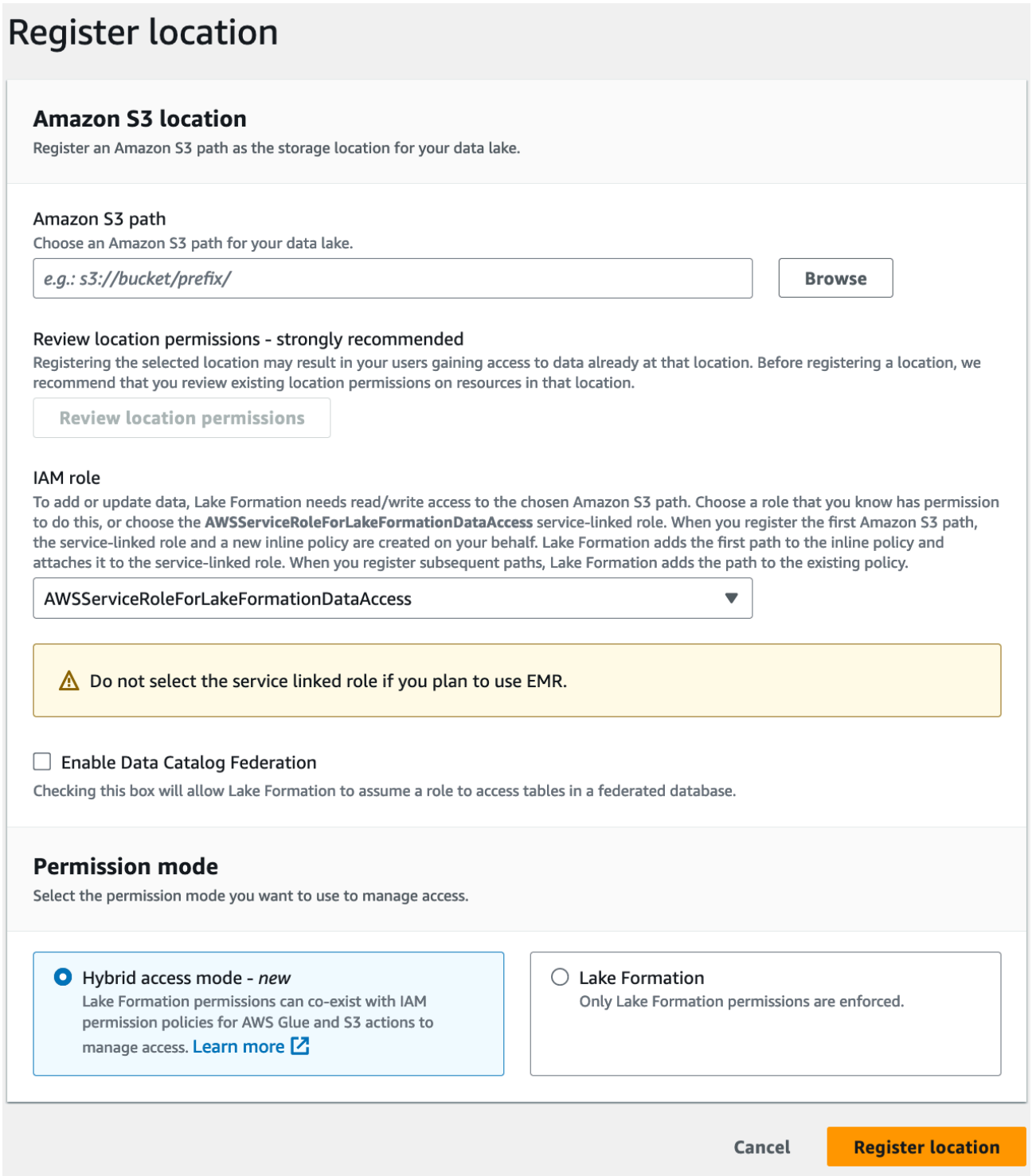
シナリオの説明 – データロケーションは、Lake Formation に登録されていません。Data Catalog データベースとテーブルへのユーザーのアクセスは、Amazon S3 および AWS Glue アクションの IAM アクセス許可ポリシーによって決定されます。
デフォルトでは、この IAMAllowedPrincipals グループにはデータベース内のすべてのテーブルに対する Super 許可があります。

Lake Formation に登録されていないデータロケーションのハイブリッドアクセスモードを有効にするには

1. Amazon S3 ロケーションを登録して、ハイブリッドアクセスモードを有効にします。

   Console

   1. Lake Formation コンソールにデータレイク管理者としてサインインします。
   2. ナビゲーションペインで、[管理] の [データレイクのロケーション] を選択します。
   3. [Register location] (ロケーションを登録) を選択します。



4. [ロケーションを登録] ウィンドウで、Lake Formation に登録する [Amazon S3] パスを選択します。
5. [IAM ロール] で、`AWSServiceRoleForLakeFormationDataAccess` サービスリンクロール (デフォルト)、または「ロケーションの登録に使用されるロールの要件」の要件を満たすカスタム IAM ロールを選択します。

6. [ハイブリッドアクセスモード] を選択すると、登録された口ケーションを指すオプトインプリンシパルと Data Catalog データベースおよびテーブルに、きめ細かい Lake Formation アクセスコントロールポリシーが適用されます。

   Lake Formation を選択すると、Lake Formation は登録された口ケーションへのアクセスリクエストを承認できるようになります。

7. [Register location] (口ケーションを登録) を選択します。

AWS CLI

次の例では、HybridAccessEnabled:true/false と設定して、Lake Formation にデータ口ケーションを登録しています。HybridAccessEnabled パラメータのデフォルト値は false です。Amazon S3 パス、ロール名、 AWS アカウント ID を有効な値に置き換えます。

```
aws lakeformation register-resource --cli-input-json file:file path
json:
    {
        "ResourceArn": "arn:aws:s3:::s3-path",
        "UseServiceLinkedRole": false,
        "RoleArn": "arn:aws:iam::<123456789012>:role/<role-name>",
        "HybridAccessEnabled": true
    }
```

2. ハイブリッドアクセスモードでリソースに Lake Formation 許可を使用するようにアクセス許可を付与し、プリンシパルをオプトインする

ハイブリッドアクセスモードでプリンシパルとリソースをオプトインする前に、ハイブリッドアクセスモードで Lake Formation に登録された場所があるデータベースとテーブルにIAMAllowedPrincipals、 Superまたは グループのAllアクセス許可が存在することを確認します。

> **Note**
>
> データベース内の All tables に IAMAllowedPrincipals グループアクセス許可を付与することはできません。ドロップダウンメニューから各テーブルを個別に選択し、アクセス許可を付与する必要があります。また、データベースに新しいテーブルを作成するときは、[データカタログの設定] で [Use only IAM access control for new tables in new databases] を選択できます。このオプションでは、データ

ベース内に新しいテーブルを作成すると、自動的に IAMAllowedPrincipals グループに Super 許可が付与されます。

Console

1. Lake Formation コンソールのデータカタログで、カタログ、データベース、またはテーブルを選択します。
2. リストからカタログ、データベース、またはテーブルを選択し、アクションメニューから付与を選択します。
3. プリンシパルを選択し、名前付きリソース方式または LF タグを使用して、データベース、テーブル、および列に対するアクセス許可を付与します。

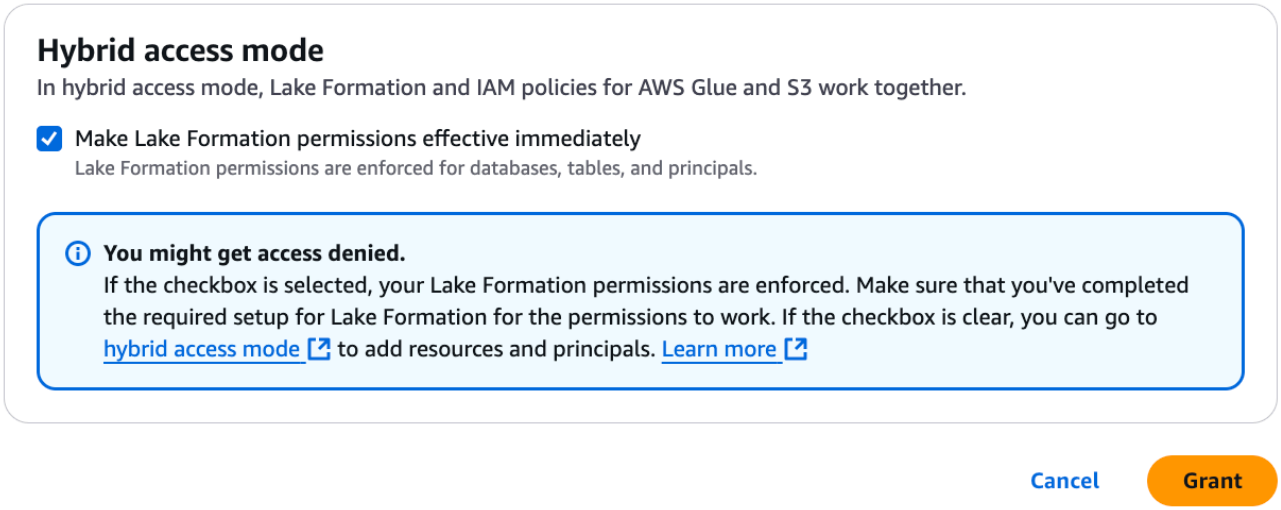
   または、データアクセス許可を選択し、リストからアクセス許可を付与するプリンシパルを選択し、付与を選択します。

   データアクセス許可の付与に関する詳細については、「データカタログリソースに対するアクセス許可の付与」を参照してください。

   > **Note**
   >
   > プリンシパルにテーブル作成のアクセス許可を付与する場合は、プリンシパルにデータロケーション許可 (DATA_LOCATION_ACCESS) を付与する必要もあります。このアクセス許可はテーブルの更新には必要ありません。
   > 詳細については、「データロケーション許可の付与」を参照してください。

4. [名前付きリソース方式] を使用してアクセス許可を付与する場合、プリンシパルとリソースをオプトインするオプションが [データ許可の付与] ページの下部に表示されます。

   プリンシパルとリソースの Lake Formation 許可を有効にするには、[Lake Formation 許可をすぐに有効にする] を選択します。



5. [付与] を選択します。

   データロケーションを指しているテーブル A のプリンシパル A をオプトインした場合、データロケーションがハイブリッドモードで登録されていれば、プリンシパル A は Lake Formation 許可を使用してこのテーブルのロケーションにアクセスできます。

AWS CLI

以下の例では、ハイブリッドアクセスモードでプリンシパルとテーブルをオプトインしています。ロール名、 AWS アカウント ID、データベース名、およびテーブル名を有効な値に置き換えます。

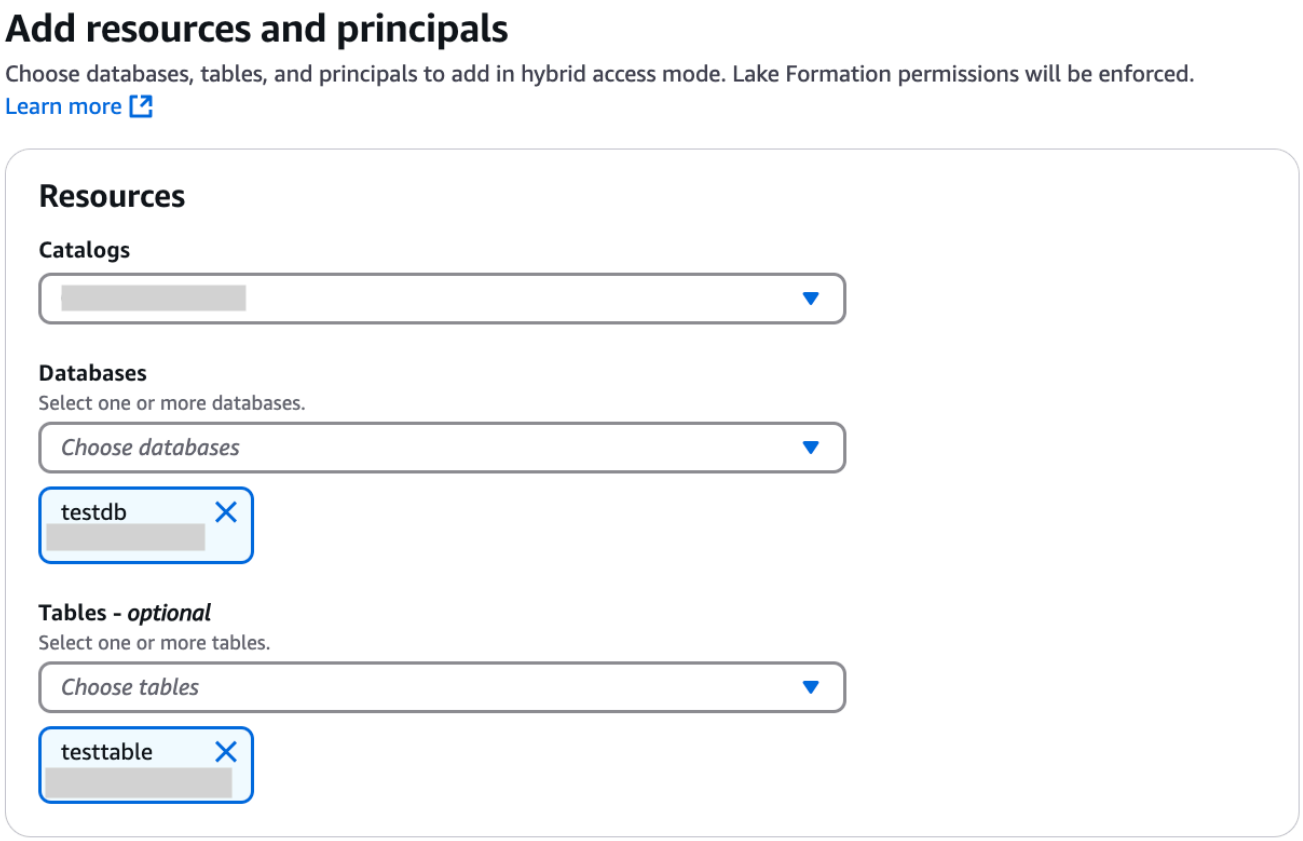
```
aws lakeformation create-lake-formation-opt-in --cli-input-json file://file path
json:
  {
        "Principal": {
            "DataLakePrincipalIdentifier":
 "arn:aws:iam::<123456789012>:role/<hybrid-access-role>"
        },
        "Resource": {
            "Table": {
                "CatalogId": "<123456789012>",
                "DatabaseName": "<hybrid_test>",
                "Name": "<hybrid_test_table>"
            }
        }
    }
```

a. (Optional) アクセス許可を付与するために LF タグを選択した場合は、別のステップで Lake Formation 許可を使用するようにプリンシパルをオプトインできます。これを行うには、左側のナビゲーションバーの [アクセス許可] で [ハイブリッドアクセスモード] を選択します。

b. [ハイブリッドアクセスモード] ページの下部にある [追加] を選択して、リソースとプリンシパルをハイブリッドアクセスモードに追加します。

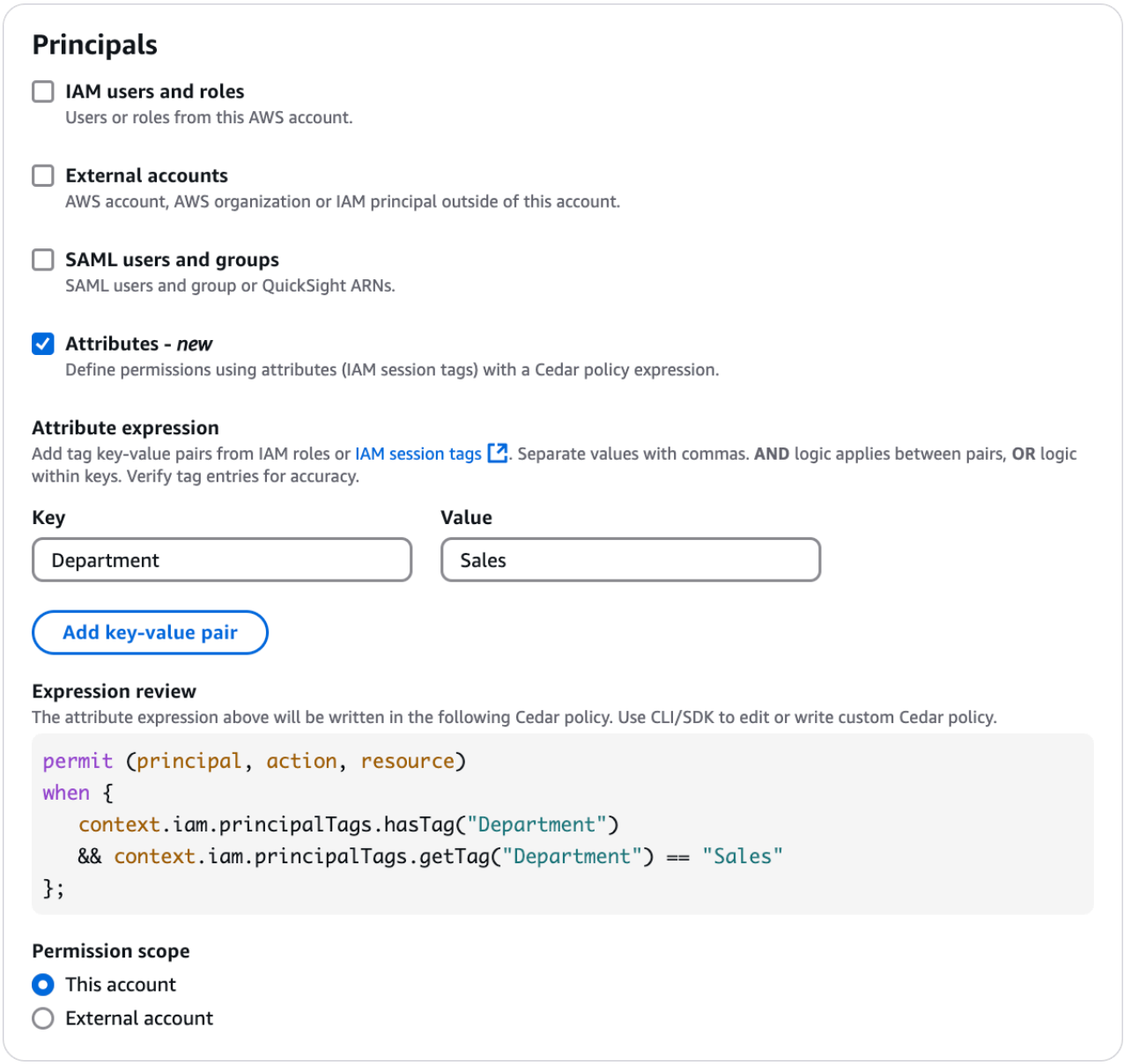
c. リソースとプリンシパルの追加ページで、ハイブリッドアクセスモードで登録されているカタログ、データベース、テーブルを選択します。

アクセスを許可するデータベースで、`All tables` を選択できます。



d. ハイブリッドアクセスモードで Lake Formation アクセス許可を使用するようにオプトインするプリンシパルを選択します。

- プリンシパル – IAM ユーザーとロールは、同じアカウントまたは別のアカウントで選択できます。SAML ユーザーとグループを選択することもできます。
- 属性 – 属性を選択して、属性に基づいてアクセス許可を付与します。



- キーと値のペアを入力して、属性に基づいて権限を作成します。コンソールで Cedar ポリシー式を確認します。Cedar の詳細については、「Cedar とは」を参照してください。| Cedar Policy Language Reference GuideLink を参照してください。
- [Add] (追加) を選択します。

  属性が一致するすべての IAM ロール/ユーザーにアクセス権が付与されます。

e. [Add] (追加) を選択します。

Lake Formation リソースをハイブリッドリソースに変換する

現在 Data Catalog データベースとテーブルに Lake Formation 許可を使用している場合は、ロケーションの登録プロパティを編集してハイブリッドアクセスモードを有効にできます。これにより、既存の Lake Formation アクセス許可を中断することなく、Amazon S3 の IAM アクセス許可ポリシーと AWS Glue アクションを使用して、新しいプリンシパルに同じリソースへのアクセスを提供できます。

シナリオの説明 – 以下のステップは、Lake Formation にデータロケーションを登録していて、そのロケーションを指すデータベース、テーブル、または列に対するプリンシパルのアクセス許可を設定していることを前提としています。そのロケーションが、サービスにリンクされたロールに登録されている場合、ロケーションパラメータを更新してハイブリッドアクセスモードを有効にすることはできません。`IAMAllowedPrincipals` グループには、データベースとそのすべてのテーブルの Super 許可がデフォルトで与えられます。

> **Important**
>
> その口ケーションのデータにアクセスするプリンシパルをオプトインすることなく、口ケーションの登録をハイブリッドアクセスモードに更新しないでください。

Lake Formation に登録されたデータロケーションのハイブリッドアクセスモードを有効にする

1.

> **Warning**
>
> 他の既存のユーザーやワークロードのアクセス許可ポリシーを中断しないようにするため、Lake Formation が管理するデータロケーションをハイブリッドアクセスモードに変換することはお勧めしません。

Lake Formation 許可を持つ既存のプリンシパルをオプトインします。

1. カタログ、データベース、テーブルのプリンシパルに付与したアクセス許可を一覧表示して確認します。詳細については、「Lake Formation でのデータベースとテーブル許可の表示」を参照してください。
2. 左側のナビゲーションバーの [アクセス許可] で [ハイブリッドアクセスモード] を選択し、[追加] を選択します。

3. プリンシパルとリソースの追加ページで、ハイブリッドアクセスモードで使用する Amazon S3 データの場所からカタログ、データベース、テーブルを選択します。既に Lake Formation 許可を持っているプリンシパルを選択します。
4. ハイブリッドアクセスモードで Lake Formation 許可を使用するようにプリンシパルをオプトインするには、[追加] を選択します。

2. [ハイブリッドアクセスモード] オプションを選択して Amazon S3 バケット/プレフィックス登録を更新します。

Console

1. Lake Formation コンソールにデータレイク管理者としてサインインします。
2. ナビゲーションペインの [Register and ingest] (登録および取り込み) で [Data lake locations] (データレイクのロケーション) を選択します。
3. ロケーションを選択し、[アクション] メニューの [削除] を選択します。
4. [ハイブリッドアクセスモード] を選択します。
5. [保存] を選択します。
6. Data Catalog で、データベースまたはテーブルを選択し、`IAMAllowedPrincipals` という仮想グループに `Super` または `All` 許可を付与します。
7. ロケーションの登録プロパティを更新したときに、既存の Lake Formation ユーザーのアクセスが中断されていないことを検証します。Lake Formation プリンシパルとして Athena コンソールにサインインし、更新されたロケーションを指すテーブルに対してサンプルクエリを実行します。

   同様に、IAM アクセス許可ポリシーを使用してデータベースとテーブルにアクセスしている AWS Glue ユーザーのアクセスを確認します。

AWS CLI

次の例では、HybridAccessEnabled:true/false と設定して、Lake Formation にデータロケーションを登録しています。`HybridAccessEnabled` パラメータのデフォルト値は false です。Amazon S3 パス、ロール名、 AWS アカウント ID を有効な値に置き換えます。

```
aws lakeformation update-resource --cli-input-json file://file path
json:
{
    "ResourceArn": "arn:aws:s3:::<s3-path>",
```

```
    "RoleArn": "arn:aws:iam::<123456789012>:role/<test>",
    "HybridAccessEnabled": true
}
```

ハイブリッドアクセスモードを使用した AWS Glue リソースの共有

既存の Data Catalog ユーザーの IAM AWS アカウント ベースのアクセスを中断することなく、別の AWS アカウント または別の のプリンシパルとデータを共有します。

シナリオの説明 - プロデューサーアカウントには、Amazon S3 の IAM プリンシパルポリシーと AWS Glue アクションを使用してアクセスが制御された Data Catalog データベースがあります。データベースのデータロケーションは、Lake Formation に登録されていません。IAMAllowedPrincipals グループには、デフォルトで、データベースとそのすべてのテーブルに対する Super アクセス許可があります。

ハイブリッドアクセスモードでクロスアカウントの Lake Formation 許可を付与する

1. プロデューサーアカウントの設定

    1. lakeformation:PutDataLakeSettings IAM アクセス許可を持つロールを使用して Lake Formation コンソールにサインインします。
    2. [データカタログの設定] ページに移動し、[クロスアカウントバージョン設定] で [Version 4] を選択します。

        現在バージョン 1 または 2 を使用している場合は、バージョン 3 への更新について、「クロスアカウントデータ共有のバージョン設定の更新」の手順を参照してください。

        バージョン 3 から 4 にアップグレードする場合、アクセス許可ポリシーを変更する必要はありません。
    3. ハイブリッドアクセスモードで共有する予定のデータベースまたはテーブルの Amazon S3 ロケーションを登録します。
    4. 上記のステップにおいて、ハイブリッドアクセスモードでデータロケーションを登録したデータベースとテーブルに、IAMAllowedPrincipals グループに対する Super 許可があることを確認します。
    5. Lake Formation のアクセス許可を AWS 組織、組織単位 (OUs) に付与するか、別のアカウントの IAM プリンシパルに直接付与します。

6. IAM プリンシパルに直接許可を付与する場合は、コンシューマーアカウントからプリンシパルにオプトインし、[Lake Formation 許可をすぐに有効にする] オプションを有効にして、ハイブリッドアクセスモードで Lake Formation 許可を適用します。

   別の AWS アカウントにクロスアカウントアクセス許可を付与する場合、アカウントをオプトインすると、Lake Formation アクセス許可はそのアカウントの管理者にのみ適用されます。受信者アカウントのデータレイク管理者は、アクセス許可をカスケードし、アカウントのプリンシパルをオプトインして、ハイブリッドアクセスモードの共有リソースに Lake Formation 許可を適用する必要があります。

   [LF タグに一致するリソース] オプションを選択してクロスアカウントアクセス許可を付与する場合は、まずアクセス許可の付与ステップを完了する必要があります。Lake Formation コンソールの左側のナビゲーションバーにある [アクセス許可] で [ハイブリッドアクセスモード] を選択することで、プリンシパルとリソースをハイブリッドアクセスモードに別のステップとしてオプトインできます。次に、[追加] を選択して、Lake Formation 許可を適用するリソースとプリンシパルを追加します。

2. コンシューマーアカウントの設定

   1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にデータレイク管理者としてサインインします。
   2. https://console.aws.amazon.com/ram/home に移動し、リソース共有の招待を受け入れます。AWS RAM コンソールの「自分と共有」タブには、アカウントと共有されているデータベースとテーブルが表示されます。
   3. Lake Formation の共有データベースまたはテーブルへのリソースリンクを作成します。
   4. リソースリンクの `Describe` 許可と (元の共有リソースの) `Grant on target` 許可を (コンシューマー) アカウントの IAM プリンシパルに付与します。
   5. 共有されているデータベースまたはテーブルの Lake Formation 許可を、アカウントのプリンシパルに付与します。[Lake Formation 許可をすぐに有効にする] オプションを有効することで、プリンシパルとリソースをオプトインし、ハイブリッドアクセスモードで Lake Formation 許可を適用します。
   6. Athena のサンプルクエリを実行して、プリンシパルの Lake Formation 許可をテストします。Amazon S3 および AWS Glue アクションの IAM プリンシパルポリシーを使用して、AWS Glue ユーザーの既存のアクセスをテストします。

(オプション) データアクセス用の Amazon S3 バケットポリシーと、Lake Formation 許可を使用するように設定したプリンシパルの AWS Glue と Amazon S3 データアクセス用の IAM プリンシパルポリシーを削除します。

ハイブリッドアクセスモードを使用して Lake Formation リソースを共有する

外部アカウントの新しい Data Catalog ユーザーが、既存の Lake Formation のクロスアカウント共有アクセス許可を中断することなく、IAM ベースのポリシーを使用して Data Catalog データベースとテーブルにアクセスできるようにします。

シナリオの説明 – プロデューサーアカウントには、アカウントレベルまたは IAM プリンシパルレベルで外部 (コンシューマー) アカウントと共有される Lake Formation 管理データベースとテーブルがあります。データベースのデータロケーションは、Lake Formation に登録されています。`IAMAllowedPrincipals` グループには、データベースとそのテーブルに対する `Super` 許可はありません。

既存の Lake Formation 許可を中断することなく、IAM ベースのポリシーを介して新しい Data Catalog ユーザーにクロスアカウントアクセスを許可する

1. プロデューサーアカウントの設定

    1. `lakeformation:PutDataLakeSettings` を持つロールを使用して Lake Formation コンソールにサインインします。
    2. [データカタログの設定] ページの [クロスアカウントバージョン設定] で、[`Version 4`] を選択します。

        現在バージョン 1 または 2 を使用している場合は、バージョン 3 への更新について、「クロスアカウントデータ共有のバージョン設定の更新」の手順を参照してください。

        バージョン 3 から 4 へのアップグレードには、アクセス許可ポリシーの変更は必要ありません。
    3. データベースとテーブルでプリンシパルに付与したアクセス許可を一覧表示します。詳細については、「Lake Formation でのデータベースとテーブル許可の表示」を参照してください。
    4. プリンシパルとリソースをオプトインすることで、既存の Lake Formation のクロスアカウントアクセス許可を再付与します。

> **Note**
>
> データロケーション登録をハイブリッドアクセスモードに更新してクロスアカウントアクセス許可を付与する前に、アカウントごとに少なくとも 1 つのクロスアカウントデータ共有を再付与する必要があります。このステップは、 AWS RAM リソース共有にアタッチされた AWS RAM 管理アクセス許可を更新するために必要です。
> 2023 年 7 月、Lake Formation はデータベースとテーブルの共有に使用される AWS RAM マネージドアクセス許可を更新しました。
>
> - `arn:aws:ram::aws:permission/AWSRAMLFEnabledGlueAllTablesReadWriteForDatabase` (データベースレベルの共有ポリシー)
> - `arn:aws:ram::aws:permission/AWSRAMLFEnabledGlueTableReadWrite` (テーブルレベルの共有ポリシー)
>
> 2023 年 7 月より前に行われたクロスアカウントアクセス許可の付与には、これらの更新された AWS RAM アクセス許可はありません。
> クロスアカウントアクセス許可をプリンシパルに直接付与した場合は、それらのアクセス許可を個別にプリンシパルに再付与する必要があります。このステップをスキップすると、共有リソースにアクセスするプリンシパルに不正な組み合わせエラーが発生する可能性があります。

5. https://console.aws.amazon.com/ram/home に移動します。
6. AWS RAM コンソールの共有元タブには、外部アカウントまたはプリンシパルと共有したデータベース名とテーブル名が表示されます。

   共有リソースにアタッチされたアクセス許可に、正しい ARN があることを確認します。
7. AWS RAM 共有内のリソースが `Associated`ステータスであることを確認します。ステータスが `Associating` と表示される場合は、`Associated` 状態になるまで待ちます。ステータスが `Failed` になった場合は、停止して Lake Formation サービスチームにご連絡ください。
8. 左側のナビゲーションバーの [アクセス許可] で [ハイブリッドアクセスモード] を選択し、[追加] を選択します。
9. [プリンシパルとリソースの追加] ページには、アクセス権のあるデータベース、テーブル、またはその両方とプリンシパルが表示されます。プリンシパルとリソースを追加または削除することで、必要な更新を行うことができます。
10. ハイブリッドアクセスモードに変更するデータベースとテーブルの Lake Formation 許可を持つプリンシパルを選択します。データベースとテーブルを選択します。

11ハイブリッドアクセスモードで Lake Formation 許可を適用するようにプリンシパルをオプトインするには、[追加] を選択します。

12.データベースと選択したテーブルの仮想グループ `IAMAllowedPrincipals` に `Super` 許可を付与します。

13Amazon S3 ロケーションの Lake Formation 登録をハイブリッドアクセスモードに編集します。

14Amazon S3 AWS Glue actions の IAM アクセス許可ポリシーを使用して、外部 (コンシューマー) アカウントの AWS Glue ユーザーにアクセス許可を付与します。

2. コンシューマーアカウントの設定

   1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にデータレイク管理者としてサインインします。
   2. https://console.aws.amazon.com/ram/home に移動し、リソース共有の招待を受け入れます。AWS RAM ページのリソース共有タブには、アカウントと共有されているデータベース名とテーブル名が表示されます。

      AWS RAM 共有の場合は、アタッチされたアクセス許可に共有 AWS RAM 招待の正しい ARN があることを確認します。 AWS RAM 共有内のリソースが `Associated`ステータスになっているかどうかを確認します。ステータスが `Associating` と表示される場合は、`Associated` 状態になるまで待ちます。ステータスが `Failed` になった場合は、停止して Lake Formation サービスチームにご連絡ください。
   3. Lake Formation の共有データベースまたはテーブルへのリソースリンクを作成します。
   4. リソースリンクの `Describe` 許可と (元の共有リソースの) `Grant on target` 許可を (コンシューマー) アカウントの IAM プリンシパルに付与します。
   5. 次に、共有データベースまたはテーブルのアカウントのプリンシパルに Lake Formation 許可を設定します。

      左側のナビゲーションバーの [アクセス許可] で、[ハイブリッドアクセスモード] を選択します。
   6. [ハイブリッドアクセスモード] ページの下部にある [追加] を選択して、プリンシパルと、プロデューサーアカウントから共有されているデータベースまたはテーブルをオプトインします。
   7. Amazon S3 AWS Glue actions の IAM アクセス許可ポリシーを使用して、アカウントの AWS Glue ユーザーにアクセス許可を付与します。

8. Athena を使用してテーブルで個別のサンプルクエリを実行して、ユーザーの Lake Formation のアクセス許可と AWS Glue アクセス許可をテストする

   (オプション) ハイブリッドアクセスモードになっているプリンシパルに対する Amazon S3 の IAM 許可ポリシーをクリーンアップします。

ハイブリッドアクセスモードからプリンシパルとリソースを削除する

以下のステップに従って、ハイブリッドアクセスモードからデータベース、テーブル、およびプリンシパルを削除します。

Console

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインします。
2. [アクセス許可] で [ハイブリッドアクセスモード] を選択します。
3. [ハイブリッドアクセスモード] ページで、データベース名またはテーブル名の横にあるチェックボックスを選択し、[Remove] を選択します。
4. 警告メッセージが表示され、アクションの確認を求められます。[を削除] を選択します。

   Lake Formation はこれらのリソースにアクセス許可を強制しなくなり、このリソースへのアクセスは IAM と アクセス AWS Glue 許可を使用して制御されます。これにより、ユーザーが適切な IAM アクセス許可を持っていないと、このリソースにアクセスできなくなる可能性があります。

AWS CLI

次の例は、ハイブリッドアクセスモードからリソースを削除する方法を示しています。

```
aws lakeformation delete-lake-formation-opt-in --cli-input-json file://file path

json:
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::<123456789012>:role/role name"
    },
    "Resource": {
        "Table": {
```

```
            "CatalogId": "<123456789012>",
            "DatabaseName": "<database name>",
            "Name": "<table name>"
          }
    }
}
```

ハイブリッドアクセスモードでプリンシパルとリソースを表示する

以下のステップに従って、ハイブリッドアクセスモードでデータベース、テーブル、プリンシパルを表示します。

Console

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインします。
2. [アクセス許可] で [ハイブリッドアクセスモード] を選択します。
3. [ハイブリッドアクセスモード] ページには、現在ハイブリッドアクセスモードになっているリソースとプリンシパルが表示されます。

AWS CLI

次の例は、ハイブリッドアクセスモードのすべてのオプトインプリンシパルとリソースを一覧表示する方法を示しています。

```
aws lakeformation list-lake-formation-opt-ins
```

次の例は、特定のプリンシパルリソースペアのオプトインを一覧表示する方法を示しています。

```
aws lakeformation list-lake-formation-opt-ins --cli-input-json file://file path

json:
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::<account-id>:role/<role name>"
```

```
    },
    "Resource": {
        "Table": {
            "CatalogId": "<account-id>",
            "DatabaseName": "<database name>",
            "Name": "<table name>"
          }
    }
}
```

その他のリソース

次のブログ記事では、IAM および Amazon S3 のアクセス許可を通じて他のユーザーが既にデータベースにアクセスできる場合に、選択したユーザーのために Lake Formation のアクセス許可をハイブリッドアクセスモードでオンボーディングする手順について説明します。アカウント内および 2 つの AWS アカウント間でハイブリッドアクセスモードを設定する手順を確認します。

- Lake Formation と IAM および Amazon S3 ポリシーを使用してアクセス AWS Glue Data Catalog を保護するための のハイブリッドアクセスモードを導入しました。 Amazon S3

でのオブジェクトの作成 AWS Glue Data Catalog

AWS Lake Formation は AWS Glue Data Catalog (データカタログ) を使用して、データレイク、データソース、変換、ターゲットに関するメタデータを保存します。メタデータは、データセット内の基になるデータに関するデータです。各 AWS アカウントには、 AWS リージョンごとに 1 つのデータカタログがあります。

データカタログ内のメタデータは、カタログ、データベース、テーブルで構成される 3 レベルのデータ階層に分類されます。さまざまなソースからのデータをカタログと呼ばれる論理コンテナにまとめます。各カタログは、Amazon Redshift データウェアハウス、 Amazon DynamoDB データベース、および Snowflake、MySQL の他 30 を超える外部データソースを含むサードパーティーデータソースソースなどのソースからのデータを表し、これらはフェデレーティッドコネクタを介して統合されています。データカタログに新しいカタログを作成して、S3 テーブルバケットまたは Redshift マネージドストレージ (RMS) にデータを保存することもできます。

テーブルには、スキーマ情報、パーティション情報、およびデータロケーションなどの基盤となるデータに関する情報が保存されます。データベースはテーブルのコレクションです。データカタロ

グには、リソースリンクも含まれています。これは、外部アカウントで共有されるカタログ、データベース、テーブルへのリンクで、データレイク内のデータへのクロスアカウントアクセスに使用されます。

データカタログは、カタログ、データベース、テーブルを含むネストされたカタログオブジェクトです。これは AWS アカウント ID によって参照され、アカウントと のデフォルトカタログです AWS リージョン。データカタログは、3 レベルの階層 (catalog.database.table) を使用してテーブルを整理します。

- カタログ – データカタログの 3 つのレベルのメタデータ階層の最上位レベル。フェデレーションを利用して、データカタログに複数のカタログを追加できます。
- データベース – テーブルとビューで構成されるメタデータ階層の 2 番目のレベル。データベースは、Amazon Redshift や Trino などの多くのデータシステムではスキーマとも呼ばれます。
- テーブルとビュー – データカタログの 3 レベルデータ階層の 3 番目のレベル。

Amazon S3 内のすべての Iceberg テーブルは、カタログ ID = AWS アカウント ID を持つデフォルトのデータカタログに保存されます。フェデレーションを使用して、Amazon Redshift、Amazon S3 Table ストレージ、またはその他のサードパーティーデータソースのテーブル定義 AWS Glue Data Catalog を保存するフェデレーションカタログを に作成できます。

トピック

- カタログの作成
- データベースを作成する
- テーブルの作成
- AWS Glue Data Catalog ビューの構築

カタログの作成

カタログは、 AWS Glue Data Catalogの 3 レベルのメタデータ階層の最上位レベルを表します。複数の方式を使用して、データをデータカタログに取り込み、マルチレベルカタログを作成できます。

外部データソースからカタログを作成する方法の詳細については、「へのデータの取り込み AWS Glue Data Catalog」を参照してください。

Lake Formation コンソールを使用してカタログを作成するには、データレイク管理者、またはカタログ作成者としてサインインしている必要があります。カタログ作成者は、Lake Formation の

CREATE_CATALOG アクセス許可を付与されているプリンシパルです。カタログ作成者のリストは、Lake Formation コンソールの [管理ロールとタスク] ページで確認することができます。このリストを表示するには、lakeformation:ListPermissions IAM 許可があり、データレイク管理者、または CREATE_CATALOG アクセス許可に対する付与オプションを持つカタログ作成者としてサインインしている必要があります。

データベースを作成する

Data Catalog のメタデータテーブルは、データベース内に保存されます。データベースは必要な数だけ作成でき、データベースごとに異なる Lake Formation 許可を付与できます。

データベースは、オプションのロケーションプロパティを持つことができます。通常、このロケーションは Lake Formation に登録されている Amazon Simple Storage Service (Amazon S3) ロケーション内にあります。ロケーションを指定するときは、プリンシパルに、データベースロケーション内のロケーションをポイントする Data Catalog テーブルを作成するためのデータロケーション許可は必要ありません。詳細については、「Underlying data access control」を参照してください。

Lake Formation コンソールを使用してデータベースを作成するには、データレイク管理者、またはデータベース作成者としてサインインしている必要があります。データベース作成者は、Lake Formation の CREATE_DATABASE 許可を付与されたプリンシパルです。データベース作成者のリストは、Lake Formation コンソールの [Administrative roles and tasks] (管理ロールとタスク) ページで確認することができます。このリストを表示するには、lakeformation:ListPermissions IAM 許可を持っており、データレイク管理者、または CREATE_DATABASE 許可に対する grant オプションを持つデータベース作成者としてサインインしている必要があります。

データベースを作成する

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、データレイク管理者またはデータベース作成者としてサインインします。
2. ナビゲーションペインの [Data catalog] で [Databases] (データベース) を選択します。
3. [Create database] (データベースを作成) を選択します。
4. [Create database] (データベースの作成) ダイアログボックスで、データベース名、オプションのロケーション、およびオプションの説明を入力します。
5. オプションで、[Use only IAM access control for new tables in this database] (このデータベース内の新しいテーブルには IAM アクセス制御のみを使用する) を選択します。

   このオプションについては、「the section called “データレイクのデフォルト設定の変更”」を参照してください。

6. [Create database] (データベースを作成) を選択します。

テーブルの作成

AWS Lake Formation メタデータテーブルには、スキーマ情報、パーティション情報、データの場所など、データレイク内のデータに関する情報が含まれています。これらのテーブルは、AWS Glue Data Catalog に保存されます。これらは、データレイクにある基盤となるデータにアクセスし、Lake Formation 許可でそのデータを管理するために使用します。テーブルは、Data Catalog 内のデータベースに保存されます。

Data Catalog テーブルを作成するには、いくつかの方法があります。

- AWS Glue でクローラを実行する。「AWS Glue デベロッパーガイド」の「クローラの定義」を参照してください。
- ワークフローを作成して実行する。「the section called “ワークフローを使用したデータのインポート”」を参照してください。
- Lake Formation コンソール、AWS Glue API、または AWS Command Line Interface (AWS CLI) を使用して、テーブルを手動で作成する。
- を使用してテーブルを作成します Amazon Athena。
- 外部アカウント内のテーブルへのリソースリンクを作成する。「the section called “リソースリンクの作成”」を参照してください。

Apache Iceberg テーブルの作成

AWS Lake Formation は、Amazon S3 に AWS Glue Data Catalog データが存在する で Apache Parquet データ形式を使用する Apache Iceberg テーブルの作成をサポートします。Data Catalog のテーブルは、データストア内のデータを表すメタデータ定義です。デフォルトでは、Lake Formation は Iceberg v2 テーブルを作成します。v1 テーブルと v2 テーブルの違いについては、Apache Iceberg ドキュメントの「形式バージョンの変更」を参照してください。

Apache Iceberg は、非常に大規模な分析データセット用のオープンテーブル形式です。Iceberg では、スキーマの変更 (スキーマ進化とも呼ばれます) を簡単に行うことができます。つまり、基になるデータを中断することなく、データテーブルの列を追加、名前変更、または削除できます。Iceberg はデータのバージョニングもサポートしているため、データの変更を経時的に追跡できます。これにより、タイムトラベル機能が有効になるため、過去のバージョンのデータにアクセスしてクエリを実行し、更新と削除の間に行われたデータの変更を分析できます。

Lake Formation コンソールまたは AWS Glue API の `CreateTable`オペレーションを使用して、データカタログに Iceberg テーブルを作成できます。詳細については、「CreateTable アクション (Python: create_table)」を参照してください。

Data Catalog に Iceberg テーブルを作成する場合、読み取りと書き込みを実行できるように、Amazon S3 でテーブル形式とメタデータファイルのパスを指定する必要があります。

Lake Formation を使用して、Amazon S3 データロケーションの登録時にきめ細かなアクセスコントロール許可を使用して Iceberg テーブルを保護できます AWS Lake Formation。Amazon S3 のソースデータおよび Lake Formation に登録されていないメタデータの場合、アクセスは Amazon S3 および AWS Glue アクションの IAM アクセス許可ポリシーによって決まります。詳細については、「Lake Formation 許可の管理」を参照してください。

> **Note**
>
> Data Catalog は、パーティションの作成と Iceberg テーブルプロパティの追加をサポートしていません。

トピック

- 前提条件
- Iceberg テーブルの作成

前提条件

Data Catalog に Iceberg テーブルを作成し、Lake Formation のデータアクセス許可を設定するには、次の要件を満たす必要があります。

1. Lake Formation にデータが登録されていない状態で Iceberg テーブルを作成するために必要なアクセス許可。

   Data Catalog にテーブルを作成するために必要なアクセス許可に加えて、テーブル作成者には次のアクセス許可が必要です。

   - リソース arn:aws:s3:::{bucketName} での `s3:PutObject`
   - リソース arn:aws:s3:::{bucketName} での `s3:GetObject`
   - リソース arn:aws:s3:::{bucketName} での `s3:DeleteObject`

2. Lake Formation にデータが登録されている状態で Iceberg テーブルを作成するために必要なアクセス許可:

   Lake Formation を使用してデータレイク内のデータを管理および保護するには、テーブルのデータを含む Amazon S3 ロケーションを Lake Formation に登録します。これは、Lake Formation が Athena、Redshift Spectrum、Amazon EMR などの AWS 分析サービスに認証情報を提供してデータにアクセスできるようにするためです。Amazon S3 ロケーションの登録の詳細については「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

   Lake Formation に登録されている、基盤となるデータを読み書きするプリンシパルには、次のアクセス許可が必要です。

   - `lakeformation:GetDataAccess`
   - `DATA_LOCATION_ACCESS`

     ロケーションに対するデータロケーション許可を持つプリンシパルは、すべての子ロケーションに対するロケーション許可も持っています。

     データロケーション許可の詳細については、「基盤となるデータのアクセスコントロール」を参照してください。

圧縮を有効にするには、Data Catalog 内のテーブルを更新するアクセス許可を持つ IAM ロールを、サービスが引き受ける必要があります。詳細については、「テーブル最適化の前提条件」を参照してください。

Iceberg テーブルの作成

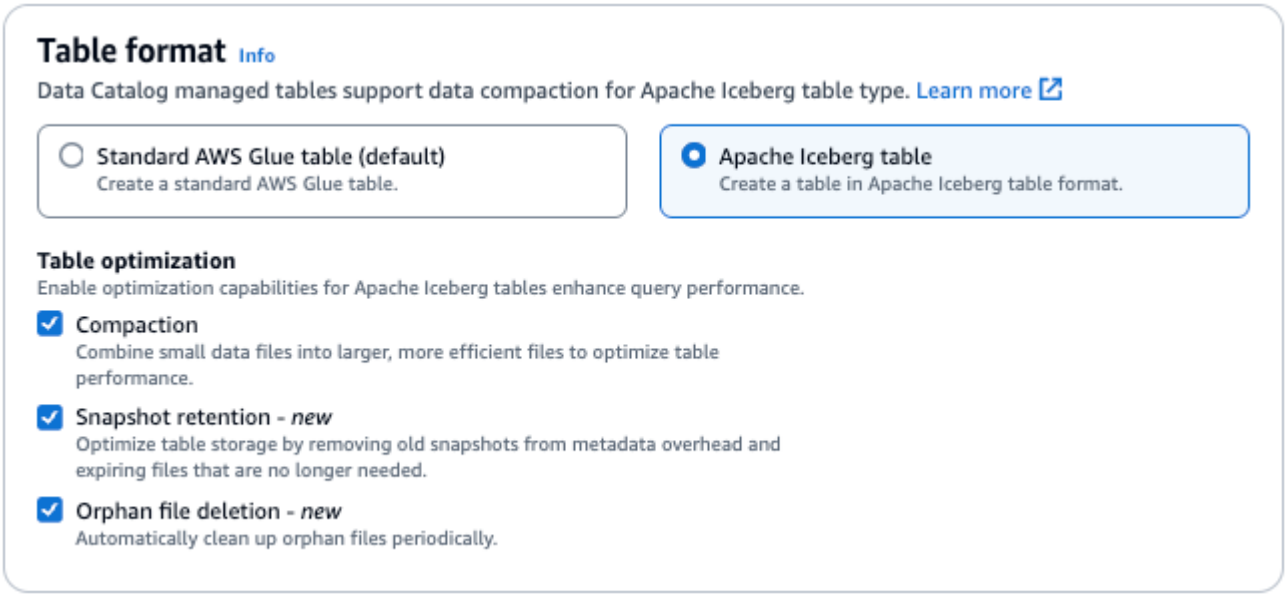
このページに記載されている AWS Command Line Interface ように、Lake Formation コンソールまたは を使用して Iceberg v1 および v2 テーブルを作成できます。 AWS Glue コンソールまたは を使用して Iceberg テーブルを作成することもできます AWS Glue クローラー。詳細については、「 AWS Glue デベロッパーガイド」の「Data Catalog とクローラー」を参照してください。

Iceberg テーブルを作成するには

Console

1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールを開きます。

2. Data Catalog で [テーブル] を選択し、[テーブルの作成] ボタンを使用して次の属性を指定します。
   - テーブル名: テーブルの名前を入力します。Athena を使用してテーブルにアクセスする場合は、「Amazon Athena ユーザーガイド」の命名に関するヒントを使用します。
   - データベース: 既存のデータベースを選択するか、新しいデータベースを作成します。
   - 説明: テーブルの説明。テーブルの内容を理解しやすくするために説明を記入できます。
   - テーブル形式: [テーブル形式] として、[Apache Iceberg] を選択します。



   - テーブル最適化
     - 圧縮 - データファイルをマージして書き換えて、古くなったデータを削除し、断片化されたデータをより大きい効率的なファイルに統合します。
     - スナップショット保持 - スナップショットは、Iceberg テーブルのタイムスタンプ付きバージョンです。スナップショット保持設定を使用すると、スナップショットを保持する期間と保持するスナップショットの数を強制できます。スナップショット保持オプティマイザーを設定すると、古い不要なスナップショットと、その基になる関連付けられたファイルを削除して、ストレージのオーバーヘッドを管理できます。
     - 孤立ファイルの削除 — 孤立ファイルは、Iceberg テーブルメタデータによって参照されなくなったファイルです。これらのファイルは、特にテーブルの削除や ETL ジョブの失敗などのオペレーションの後、時間の経過と共に蓄積される可能性があります。孤立ファイルの削除を有効にすると AWS Glue 、 はこれらの不要なファイルを定期的に識別して削除し、ストレージを解放できます。

     詳細については、「Iceberg テーブルの最適化」を参照してください。

- IAM ロール: 圧縮を実行する場合、サービスはユーザーに代わって IAM ロールを引き受けます。IAM ロールは、ドロップダウンを使用して選択できます。圧縮を有効にするために必要なアクセス許可がロールにあることを確認します。

  必要なアクセス許可の詳細については、「テーブル最適化の前提条件」を参照してください。

- ロケーション: メタデータテーブルを保存する Amazon S3 内のフォルダへのパスを指定します。Iceberg が読み取りと書き込みを実行するには、メタデータファイルと Data Catalog 内のロケーションが必要です。

- スキーマ: [列を追加] を選択して、列と、列のデータ型を追加します。空のテーブルを作成して、後でスキーマを更新することもできます。Data Catalog は Hive データ型をサポートしています。詳細については、「Hive データ型」を参照してください。

  Iceberg では、テーブルを作成した後でスキーマとパーティションを進化させることができます。[Athena クエリ] を使用してテーブルスキーマを更新し、[Spark クエリ] を使用してパーティションを更新できます。

AWS CLI

```
aws glue create-table \
    --database-name iceberg-db \
    --region us-west-2 \
    --open-table-format-input '{
      "IcebergInput": {
           "MetadataOperation": "CREATE",
           "Version": "2"
        }
      }' \
    --table-input '{"Name":"test-iceberg-input-demo",
           "TableType": "EXTERNAL_TABLE",
           "StorageDescriptor":{
              "Columns":[
                  {"Name":"col1", "Type":"int"},
                  {"Name":"col2", "Type":"int"},
                  {"Name":"col3", "Type":"string"}
               ],
              "Location":"s3://DOC_EXAMPLE_BUCKET_ICEBERG/"
           }
       }'
```

Iceberg テーブルの最適化

Lake Formation は、 AWS 分析エンジンと ETL ジョブで使用される Apache Iceberg テーブルの管理とパフォーマンスを向上させるために、複数のテーブル最適化オプションをサポートしています。これらのオプティマイザーは、効率的なストレージの使用量、クエリパフォーマンスの向上、効果的なデータ管理を実現します。Lake Formation では、次の 3 種類のテーブルオプティマイザーを使用できます。

- 圧縮 - データ圧縮では、小さいデータファイルを圧縮してストレージの使用量を減らし、読み取りパフォーマンスを向上させます。古いデータを削除して、フラグメント化されたデータをより大規模で効率的なファイルに統合するために、データファイルはマージされ、書き換えられます。圧縮は、必要に応じて自動または手動でトリガーするように設定できます。
- スナップショット保持 - スナップショットは、Iceberg テーブルのタイムスタンプ付きバージョンです。スナップショット保持設定を使用すると、スナップショットを保持する期間と保持するスナップショットの数を強制できます。スナップショット保持オプティマイザーを設定すると、古い不要なスナップショットと、その基になる関連付けられたファイルを削除して、ストレージのオーバーヘッドを管理できます。
- 孤立ファイルの削除 — 孤立ファイルは、Iceberg テーブルメタデータによって参照されなくなったファイルです。これらのファイルは、特にテーブルの削除や ETL ジョブの失敗などのオペレーションの後、時間の経過と共に蓄積される可能性があります。孤立ファイルの削除を有効にすると AWS Glue 、 はこれらの不要なファイルを定期的に識別して削除し、ストレージを解放できます。

AWS Glue コンソール、または AWS Glue API オペレーションを使用して、データカタログ内の個々の Iceberg テーブルの圧縮、スナップショット保持 AWS CLI、孤立ファイル削除オプティマイザを有効または無効にできます。

詳細については、「 AWS Glue デベロッパーガイド」の「Iceberg テーブルの最適化」を参照してください。

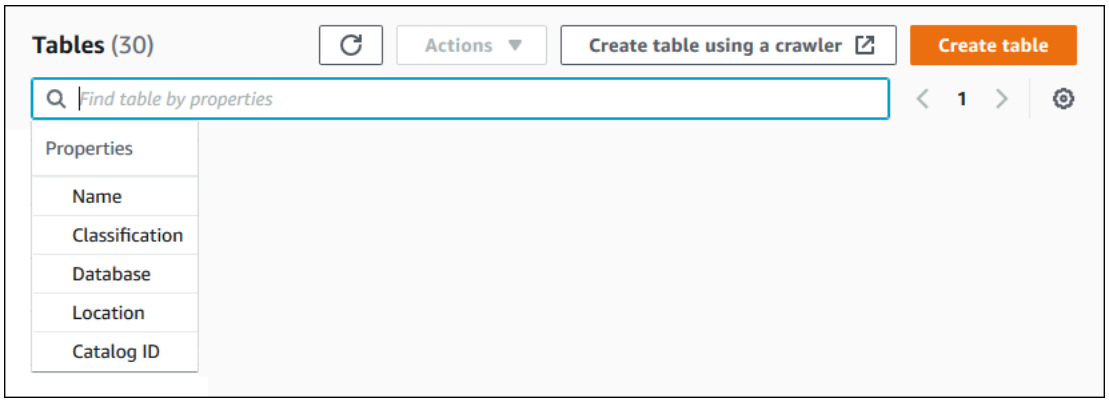
テーブルの検索

AWS Lake Formation コンソールを使用して、名前、場所、データベースを含むデータカタログテーブルなどでデータカタログテーブルを検索できます。検索結果には、Lake Formation 許可を持つテーブルのみが表示されます。

テーブルを検索する (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/lakeformation/ で Lake Formation コンソールを開きます。
2. ナビゲーションペインで、[Table] (テーブル) を選択します。
3. 検索フィールドにカーソルを置きます。このフィールドには、[プロパティでテーブルを検索] というプレースホルダテキストが表示されています。

   検索に使用できるさまざまなテーブルプロパティを示す [Properties] (プロパティ) メニューが表示されます。



4. 以下のいずれかを実行します。

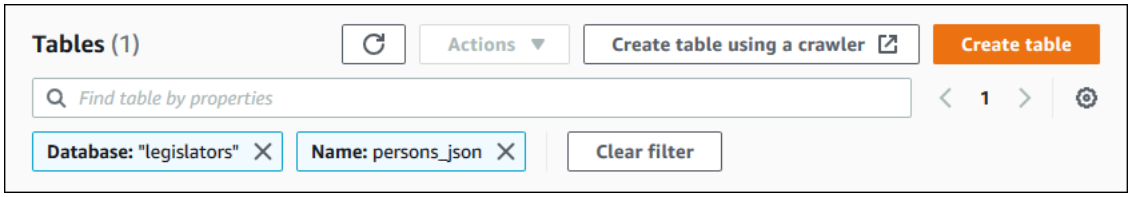
   - テーブルが含まれるデータベースで検索します。

     1. [Properties] (プロパティ) メニューから [Databases] (データベース) を選択し、表示される [Databases] (データベース) メニューからデータベースを選択するか、データベース名を入力して [Enter] キーを押します。

        データベースにある、許可を持っているテーブルがリストされます。

     2. (オプション) このリストをデータベース内の単一のテーブルに絞り込むには、もう 1 度検索フィールドにカーソルを置き、[Properties] (プロパティ) メニューから [Name] (名前) を選択して、表示される [Tables] (テーブル) メニューからテーブル名を選択するか、テーブル名を入力して [Enter] キーを押します。

        単一のテーブルがリストされ、検索フィールドの下にデータベース名とテーブル名の両方がタイルとして表示されます。



フィルターを調整するには、どちらかのタイルを閉じるか、[Clear filter] (フィルターをクリア) を選択します。

- 他のプロパティで検索します。

  1. [Properties] (プロパティ) メニューから検索プロパティを選択します。

     AWS アカウント ID で検索するには、プロパティメニューからカタログ ID を選択し、有効な AWS アカウント ID (111122223333 など) を入力し、Enter キーを押します。

     ロケーションで検索するには、[Properties] (プロパティ) メニューから [Location] (ロケーション) を選択し、表示される [Location] (ロケーション) メニューからロケーションを選択します。選択したロケーション (Amazon S3 など) のルートロケーションにあるすべてのテーブルが返されます。

を使用したテーブルの検索 AWS CLI

- 次の例は、部分検索を実行する方法を示しています。`--search-text` パラメータを使用すると、指定されたテキストがメタデータに含まれるテーブルを検索できます。この場合、名前、説明、またはその他のメタデータフィールドに「顧客」が含まれるすべてのテーブルが返されます。

```
aws glue search-tables
      --search-text "customer"
      --region AWS #####
      --max-results 10
      --sort-criteria "FieldName=Name,Sort=ASC"
```

AWS アカウント間でのデータカタログテーブルとデータベースの共有

リソースに対する Lake Formation アクセス許可を外部 AWS アカウントに付与することで、Data Catalog リソース (データベースとテーブル) を外部アカウントと共有できます。ユーザーはその後、複数のアカウントにまたがるテーブルを結合してクエリするクエリとジョブを実行できるようになります。制限はいくつかありますが、Data Catalog リソースを別のアカウントと共有する場合、その

アカウント内のプリンシパルは、そのリソースをプリンシパルの Data Catalog 内にあるかのように操作することができます。

リソースは外部 AWS アカウントの特定のプリンシパルと共有しません。リソースは AWS アカウントまたは組織と共有します。 AWS 組織とリソースを共有する場合は、その組織にあるすべてのレベルのすべてのアカウントとリソースを共有することになります。共有後、各外部アカウントのデータレイク管理者が、そのアカウント内のプリンシパルに共有リソースに対する許可を付与する必要があります。

詳細については、「Lake Formation でのクロスアカウントデータ共有」および「データカタログリソースに対するアクセス許可の付与」を参照してください。

以下も参照してください。

- 共有 Data Catalog テーブルとデータベースへのアクセスと表示
- 前提条件

AWS Glue Data Catalog ビューの構築

では AWS Glue Data Catalog、ビューは、1 つ以上のテーブルを参照する SQL クエリによってコンテンツが定義される仮想テーブルです。Amazon Athena、Amazon Redshift、または Apache Spark の SQL エディタを使用して、EMR Serverless または AWS Glue バージョン 5.0 を使用して、最大 10 個のテーブルを参照するデータカタログビューを作成できます。ビューの基盤となる参照テーブルは、同じデータベースに属することも、同じデータカタログ内の異なるデータベースに属する AWS アカウントこともできます。

Apache Hudi、Linux Foundation Delta Lakehttps://delta.io/、Apache Iceberg などのオープンテーブル形式 (OTF) の標準 AWS Glue テーブルとテーブルを参照できます。基盤となるデータは、Amazon S3 ロケーションに登録されています AWS Lake Formation。また、Lake Formation と共有された Amazon Redshift データ共有のフェデレーションテーブルからビューを作成することもできます。

データカタログビューと他のビュータイプとの区別

データカタログビューは、Apache Hive、Apache Spark、Amazon Athena のビューとは異なります。データカタログビューは のネイティブ機能であり AWS Glue Data Catalog、マルチダイアレクト定義によって作成されたビューです。Athena や Amazon Redshift Spectrum など、サポート

されている分析サービスのいずれかを使用してデータカタログビューを作成し、サポートされている他の分析サービスを使用して同じビューにアクセスできます。一方、Apache Hive、Apache Spark、Athena のビューは、Athena や Amazon Redshift などの各分析サービスで個別に作成され、そのサービス内でのみ表示およびアクセスできます。

定義者ビューとは

定義者ビューとは、そのビューを作成したプリンシパルのアクセス許可に基づいて動作する SQL ビューです。定義者ロールは、参照されるテーブルへのアクセスに必要なアクセス許可を持ち、ビューを定義する SQL ステートメントを実行します。定義者はビューを作成し、 AWS Lake Formationのきめ細かなアクセスコントロールを通じて他のユーザーと共有します。

ユーザーが定義者ビューにクエリを実行すると、クエリエンジンは、定義者ロールのアクセス許可を使用して基盤の参照テーブルにアクセスします。このアプローチにより、ユーザーはソーステーブルへの直接アクセスを必要とせずにビューを操作できるため、セキュリティが強化され、データのアクセス管理が簡素化されます。

定義者ビューを設定するには、定義者 IAM ロールをベーステーブルと同じ AWS アカウント内、またはクロスアカウント定義ロールを使用する別のアカウント内に配置できます。定義者ロールに必要なアクセス許可の詳細については、「ビュー作成の前提条件」を参照してください。

マルチダイアレクトビューのフレームワーク

データカタログは、複数の構造化クエリ言語 (SQL) ダイアレクトを使用したビューの作成をサポートしています。SQL は、リレーショナルデータベースに情報を保存および処理するために使用される言語であり、各 AWS 分析エンジンは独自のバリエーションの SQL または SQL ダイアレクトを使用します。

サポートされている分析クエリエンジンのいずれかを使用して、1 つの SQL ダイアレクトでデータカタログビューを作成します。その後、サポートされている他の分析エンジン内の別の SQL ダイアレクトで、`ALTER VIEW` ステートメントを使用してビューを更新できます。ただし、各ダイアレクトは同じテーブル、列、データ型のセットを参照する必要があります。

`GetTable` API AWS CLI とコンソールを使用して、ビューで使用できる複数のダイアレクトにアクセスできます AWS 。このように、データカタログビューには、サポートされているさまざまな分析エンジンから参照やクエリを行うことができます。

データカタログビューでは、複数のエンジンからクエリできる共通のビュースキーマとメタデータオブジェクトを定義することで、データレイク全体で統一されたビューを使用できます。

各ダイアレクトでスキーマがどのように解決されるかの詳細については、API リファレンスへのリンクを参照してください。さまざまなタイプのマッチングルールの詳細については、API ドキュメントの関連するセクションへのリンクを参照してください。

Lake Formation アクセス許可との統合

を使用して AWS Lake Formation 、ユーザーの AWS Glue Data Catalog ビューに対するアクセス許可管理を一元化できます。名前付きリソースメソッドまたは LF タグを使用して、Data Catalog ビューにきめ細かなアクセス許可を付与し、組織 AWS アカウント、 AWS 組織単位間で共有できます。リソースリンク AWS リージョン を使用して、 間で Data Catalog ビューを共有してアクセスすることもできます。これにより、ユーザーはデータソースを複製せずにデータアクセスを提供し、基になるテーブルを共有できます。

データカタログビューの `CREATE VIEW`DDL ステートメントは、Hudi、Delta Lake、Iceberg などのオープンテーブル形式 (OTF) の標準 AWS Glue テーブルとテーブルを、Lake Formation に登録されている Amazon S3 ロケーションに保存されている基盤となるデータとともに、Lake Formation と共有されている Amazon Redshift データ共有のフェデレーティッドテーブルで参照できます。テーブルのファイル形式は、ビューのクエリに使用されるエンジンがサポートしている形式であれば、任意の形式にすることができます。また、実行されているエンジンの組み込み関数を参照することもできますが、他のエンジン固有のリソースは許可されない場合があります。詳細については、「データカタログビューの考慮事項と制限」を参照してください。

ユースケース

データカタログビューの重要なユースケースを以下に示します。

- 1 つのビュースキーマでアクセス許可を作成および管理します。これにより、複数のエンジンで作成された重複したビューに対するアクセス許可に整合性がなくなるリスクを回避できます。
- 基になる参照テーブルに直接アクセス許可を付与しなくても、複数のテーブルを参照するビューに対するアクセス許可をユーザーに付与できます。
- LF タグを使用してテーブルの行レベルのフィルタリングを実現します (LF タグのカスケードは列レベルまでに限られます)。これは、ビューに LF タグを適用し、LF タグベースのアクセス許可をユーザーに付与することで行います。

ビューでサポートされている AWS 分析サービス

次の AWS 分析サービスは、データカタログビューの作成をサポートしています。

- Amazon Redshift

- Amazon Athena バージョン 3
- EMR Serverless 上の Apache Spark
- AWS Glue バージョン 5.0 の Apache Spark

その他のリソース

データカタログの詳細については、このガイドおよび以下のリソースを参照してください。

次のビデオでは、Athena と Amazon Redshift からビューを作成してクエリを実行する方法を説明しています。

トピック

- ビュー作成の前提条件
- DDL ステートメントを使用したデータカタログビューの作成
- AWS Glue APIs を使用したデータカタログビューの作成
- データカタログビューに対する許可の付与
- マテリアライズドビュー

ビュー作成の前提条件

- データカタログでビューを作成するには、参照テーブルの基礎となる Amazon S3 データの場所を Lake Formation に登録する必要があります。Lake Formation へのデータの登録の詳細については、「データレイクへの Amazon S3 ロケーションの追加」を参照してください。
- データカタログビューを作成できるのは IAM ロールだけです。他の IAM ID はデータカタログビューを作成できません。
- ビューを定義する IAM ロールには、次のアクセス許可が必要です。
  - すべての列を含む、すべての参照テーブルに対する `Grantable` オプション付きの Lake Formation `SELECT` アクセス許可。
  - ビューが作成されるターゲットデータベースに対する Lake Formation `CREATE_TABLE` アクセス許可。
  - Lake Formation と AWS Glue のサービスがロールを引き受ける信頼ポリシー。

    JSON

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

```
    "Statement": [
        {
            "Sid": "DataCatalogViewDefinerAssumeRole1",
            "Effect": "Allow",
            "Principal": {
               "Service": [
                    "glue.amazonaws.com",
                    "lakeformation.amazonaws.com"
                 ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

- AWS Glue および Lake Formation の iam:PassRole アクセス許可。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DataCatalogViewDefinerPassRole1",
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "glue.amazonaws.com",
                        "lakeformation.amazonaws.com"
                      ]
                }
            }
        }
    ]
}
```

- AWS Glue および Lake Formation のアクセス許可。

JSON

```
{
    "Version":"2012-10-17",
                "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "Glue:GetDatabase",
                "Glue:GetDatabases",
                "Glue:CreateTable",
                "Glue:GetTable",
                "Glue:GetTables",
                "Glue:BatchGetPartition",
                "Glue:GetPartitions",
                "Glue:GetPartition",
                "Glue:GetTableVersion",
                "Glue:GetTableVersions",
    "Glue:PassConnection",
                "lakeFormation:GetDataAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

- `IAMAllowedPrincipals` グループに `Super` または `ALL` アクセス許可が付与されているデータベースにビューを作成することはできません。データベースに設定されている `IAMAllowedPrincipals` グループの `Super` アクセス許可を取り消すか、「ステップ 4: データストアを Lake Formation 許可モデルに切り替える」を参照するか、または [新しく作成されたテーブルのデフォルトのアクセス許可] の [このデータベースの新しいテーブルに IAM アクセスコントロールのみを使用] ボックスをオフにして新しいデータベースを作成できます。

DDL ステートメントを使用したデータカタログビューの作成

Athena、Amazon Redshift の SQL エディタと AWS Glue APIs/ を使用して AWS Glue Data Catalog ビューを作成できますAWS CLI。

SQL エディタを使用してデータカタログビューを作成するには、Athena または Redshift Spectrum を選択し、`CREATE VIEW` データ定義言語 (DDL) ステートメントを使用してビューを作成します。

最初のエンジンのダイアレクトでビューを作成した後、2 番目のエンジンの ALTER VIEW DDL ステートメントを使用してダイアレクトを追加できます。

ビューを定義するときは、次の点を考慮することが重要です。

- マルチダイアレクトビューの定義 - 複数のダイアレクトでビューを定義する場合、異なるダイアレクトのスキーマが一致している必要があります。各 SQL ダイアレクトは構文の仕様が若干異なります。データカタログビューを定義するクエリ構文は、どのダアレクトでもまったく同じ列リストに解決され、各列のタイプと名前も一致する必要があります。この情報はビューの StorageDescriptor に格納されます。各ダイアレクトでは、データカタログから、基になる同じテーブルオブジェクトを参照する必要もあります。

  DDL を使用してビューに別のダイアレクトを追加するには、ALTER VIEW ステートメントを使用できます。ALTER VIEW ステートメントでビュー定義を更新しようとすると (ストレージ記述子やビューの基になるテーブルを変更しようとした場合など)、ステートメントから「Input and existing storage descriptor mismatch」というエラーが出力されます。ビューの列タイプを確実に一致させるには、SQL のキャスト操作を使用できます。
- ビューの更新 - ビューを更新するには、UpdateTable API を使用できます。ストレージ記述子や参照テーブルを一致させずにビューを更新する場合は、FORCE フラグを指定できます (構文についてはエンジン SQL ドキュメントを参照してください)。強制更新後、ビューには強制された StorageDescriptor と参照テーブルが反映されます。それ以降の ALTER VIEW DDL は、変更された値と一致する必要があります。更新の結果として互換性のないダイアレクトが含まれるビューは、「Stale」ステータスになります。ビューのステータスは、Lake Formation コンソールおよび GetTable オペレーションを使用して確認できます。
- varchar 列タイプの文字列としての参照 - Redshift Spectrum の varchar 列タイプを文字列にキャストすることはできません。Redshift Spectrum で varchar 列タイプを持つビューが作成され、後続のダイアレクトがそのフィールドを文字列として参照しようとすると、データカタログは FORCE フラグがなくてもそのフィールドを文字列として扱います。
- 複合タイプのフィールドの処理 - Amazon Redshift ではすべての複合タイプが SUPER タイプとして扱われますが、Athena では複合タイプが指定されます。ビューに SUPER タイプのフィールドがあり、別のエンジンがその列を構造体 (<street_address:struct<street_number:int, street_name:string, street_type:string>>) などの特定の複合タイプとして参照する場合、データカタログはフィールドが特定の複合タイプであると想定し、Force フラグがなくてもそのタイプをストレージ記述子で使用します。

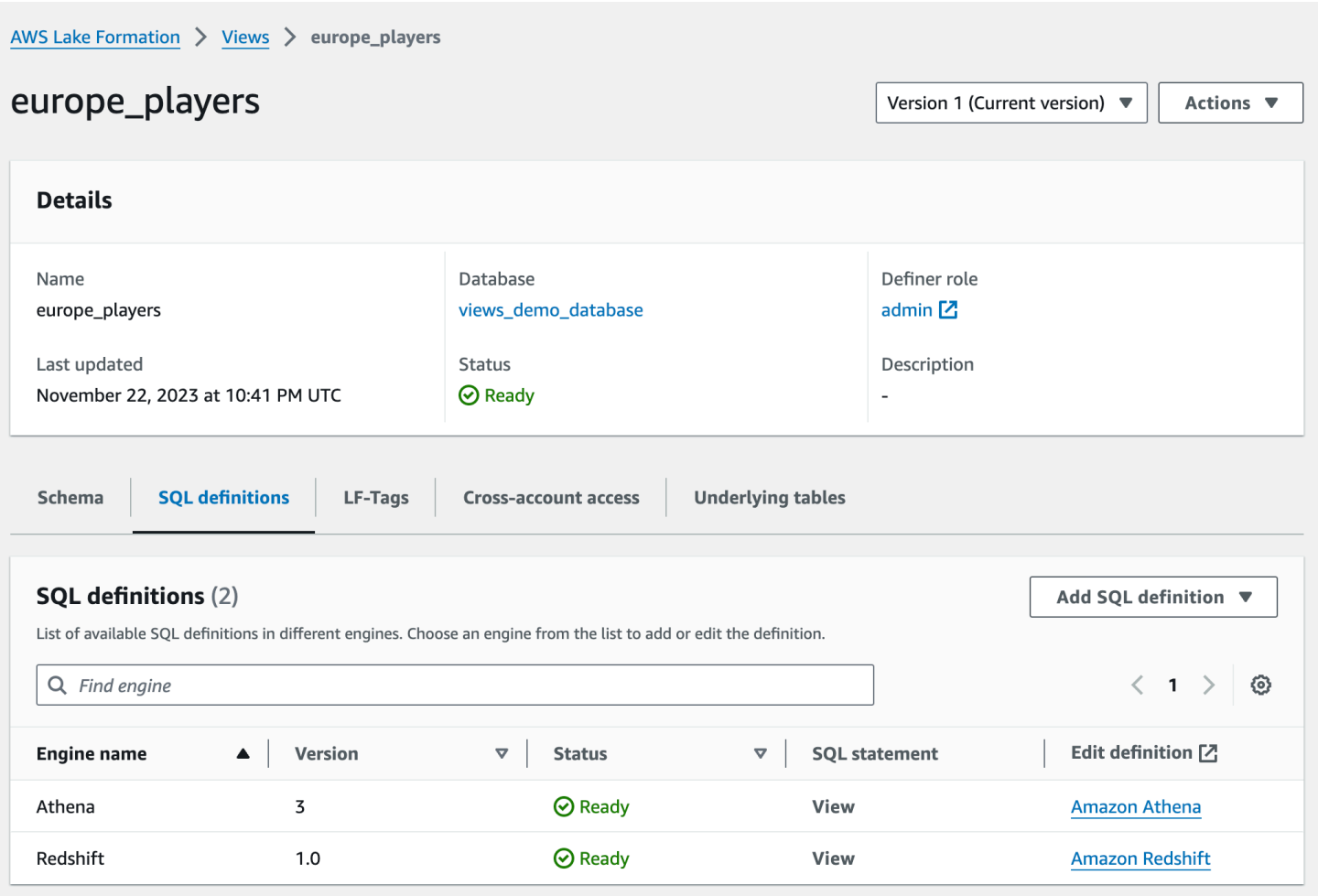
データカタログビューを作成および管理するための構文の詳細については、以下を参照してください。

- Amazon Athena ユーザーガイドの「AWS Glue Data Catalog ビューの使用」。
- 「Amazon Athena ユーザーガイド」の「Glue データカタログビューのクエリ構文」。
- Amazon Redshift データベース開発者ガイドの「AWS Glue Data Catalogでのビューの作成」。

  データカタログ内のビューに関連する SQL コマンドの詳細については、「外部ビューの作成」、「外部ビューの変更」、および「外部ビューの削除」を参照してください。

データカタログビューを作成すると、ビューの詳細が Lake Formation コンソールに表示されます。

1. Lake Formation コンソールの [データカタログ] で [ビュー] を選択します。
2. 使用可能なビューのリストがビューページに表示されます。
3. リストからビューを選択すると、詳細ページにビューの属性が表示されます。



Schema

Column 行を選択し、[LF タグの編集] を選択して、タグ値の更新や新しい LF タグの割り当てを行います。

SQL 定義

使用可能な SQL 定義のリストが表示されます。[SQL 定義を追加] を選択し、クエリエンジンを選択して SQL 定義を追加します。Edit definition 列の下にあるクエリエンジン (Athena または Amazon Redshift) を選択して、SQL 定義を更新します。

LF タグ

[LF タグを編集] を選択して、タグの値を編集したり、新しいタグを割り当てたりします。LF タグを使用すると、ビューに許可を付与できます。

クロスアカウントアクセス

データカタログビューを共有した AWS アカウント組織、組織単位 (OUs) のリストを確認できます。

基礎となるテーブル

ビューの作成に使用された SQL 定義で参照される基礎となるテーブルがこのタブに表示されます。

AWS Glue APIs を使用したデータカタログビューの作成

AWS Glue CreateTable API と UpdateTable APIs を使用して、データカタログでビューを作成および更新できます。CreateTable および UpdateTable オペレーションには、ViewDefinition を含む新しい TableInput 構造が用意されています。SearchTables、GetTable、GetTables、GetTableVersion、GetTableVersions オペレーションでは、ビューの出力構文に ViewDefinition が含められます。さらに、GetTable API の出力には新しい Status フィールドがあります。

サポートされている各クエリエンジンと Amazon Athena Amazon Redshift の SQL ダイアレクトを検証するために、2 つの新しい AWS Glue 接続を使用できます。

CreateTable および UpdateTable API は、ビューで使用する場合は非同期です。これらの API が複数の SQL ダイアレクトで呼び出されると、各エンジンで呼び出しが検証され、そのエンジンでダイアレクトを実行できるかどうか、および各ダイアレクトから返される結果のビューのスキーマが一致するかどうかが判定されます。この AWS Glue サービスは、これらの接続を使用して、分析

エンジンへの内部呼び出しを行います。これらの呼び出しは、CREATE VIEW または ALTER VIEW SQL DDL がエンジンで実行されたとした場合にそのエンジンで行われる検証をシミュレートします。

指定された SQL が有効で、各ビューダイアレクトのスキーマが一致すれば、 AWS Glue API は結果を不可分的にコミットします。この不可分性により、複数のダイアレクトを持つビューをダウンタイムなしで作成または変更できます。

トピック

- ステータスを検証するための AWS Glue 接続の作成
- ビューの生成ステータスの検証
- 非同期状態とオペレーション
- 非同期オペレーションでの作成失敗シナリオの例

ステータスを検証するための AWS Glue 接続の作成

CreateTable または UpdateTableオペレーションを使用して AWS Glue Data Catalog ビューを作成または更新するには、検証用の新しいタイプの AWS Glue 接続を作成し、サポートされている分析エンジンに提供する必要があります。この接続は、Athena または Amazon Redshift でデータカタログビューを使用するために必要です。これらの接続は AWS CLI、、 AWS SDKs、または AWS Glue APIs を使用してのみ作成できます。を使用して AWS Glue 接続 AWS マネジメントコンソールを作成することはできません。

> **Note**
>
> ビュー定義者のロールと CreateTable または UpdateTable を呼び出すロールが異なる場合は、両方の IAM ポリシーステートメントに glue:PassConnection アクセス許可が必要です。

詳細については、「create-connection AWS CLI ドキュメント」を参照してください。

AWS CLI 接続を作成するための コマンド

接続を作成するための AWS CLI コマンドを次に示します。

```
aws glue create-connection --region us-east-1
--endpoint-url https://glue.us-east-1.amazonaws.com
```

```
--cli-input-json file:///root/path/to/create-connection.json
```

AWS CLI 入力 JSON

Amazon Redshift の場合:

```
{
    "CatalogId": "123456789012",
    "ConnectionInput": {
        "ConnectionType": "VIEW_VALIDATION_REDSHIFT",
        "Name": "views-preview-cluster-connection-2",
        "Description": "My first Amazon Redshift validation connection",
        "ConnectionProperties": {
            "DATABASE": "dev",
            "CLUSTER_IDENTIFIER": "glue-data-catalog-views-preview-cluster"
        }
    }
}
```

Amazon Athena の場合:

```
{
    "CatalogId": "123456789012",
    "ConnectionInput": {
        "ConnectionType": "VIEW_VALIDATION_ATHENA",
        "Name": "views-preview-cluster-connection-3",
        "Description": "My first Amazon Athena validation connection",
        "ConnectionProperties": {
            "WORKGROUP_NAME": "workgroup-name"
        }
    }
}
```

ビューの生成ステータスの検証

CreateTable または UpdateTable オペレーションを実行すると、GetTable API 出力の Status フィールドに、ビューの作成ステータスの詳細が示されます。テーブルがまだ存在しないcreateリクエストの場合、 は非同期プロセス中に空のテーブル AWS Glue を作成します。GetTable の呼び出し時には、オプションでブール値の IncludeStatusDetails フラグを渡すことができます。このフラグは、リクエストに関する診断情報を出力するように指定します。エラーが発生すると、このフラグは、各ダイアレクトの個々のステータスを含むエラーメッセージを提供します。

ビューの作成、読み取り、更新、削除 (CRUD) オペレーション中のエラーは、 AWS Glue/Lake Formation サービスでの処理中、または Amazon Redshift または Athena でのビュー SQL 検証中に発生する可能性があります。エンジンの検証中にエラーが発生した場合、 AWS Glue サービスは、エンジンから返されたエラーメッセージを提供します。

ステータスフィールド

以下はステータスフィールドです。

- Status - さまざまなジョブのタイプに依存しない汎用のステータスです。
  - QUEUED
  - IN_PROGRESS
  - SUCCESS
  - FAILED
- Action - テーブルで呼び出されたアクションを示します。現時点では、CREATE または UPDATE オペレーションのみが利用可能です。

  ビューを操作するときは、UPDATE オペレーションと CREATE オペレーションを区別することが重要です。オペレーションタイプによって、テーブルのクエリをどのように進めるかべきが決まります。

  UPDATE オペレーションは、テーブルがデータカタログに既に存在することを意味します。この場合、以前に作成されたテーブルへのクエリを問題なく続行できます。一方、CREATE オペレーションは、これまでテーブルが正常に作成されたことはないことを示します。テーブルが CREATE としてマークされている場合、そのテーブルはまだシステムに存在しないため、クエリを試みても失敗します。したがって、テーブルのクエリを試みる前にオペレーションタイプ (UPDATE または CREATE) を特定する必要があります。
- RequestedBy - 非同期の変更をリクエストしたユーザーの ARN。
- UpdatedBy - キャンセルや変更のリクエストなど、非同期変更プロセスを最後に手動で変更したユーザーの ARN。
- Error - このフィールドは状態が FAILED の場合にのみ存在します。これは親レベルの例外メッセージです。ダイアレクトごとに異なるエラーが発生する場合があります。
  - ErrorCode - 例外のタイプ。
  - ErrorMessage - 例外の簡単な説明。
- RequestTime - 変更が開始された時刻を示す ISO 8601 形式の日付文字列。
- UpdateTime - 状態が最後に更新された時刻を示す ISO 8601 形式の日付文字列。

非同期状態とオペレーション

glue:CreateTable リクエストを実行すると、データカタログビューの非同期作成が開始されます。以下のセクションでは、このドキュメントで、glue:GetTableレスポンスで使用できる AWS Glue ビューStatusの について説明します。簡潔にするために、このセクションではレスポンス全体は省略しています。

```
{
    "Table": {
        ...
        "Status": {
            ...
            "Action": "CREATE",
            "State": "QUEUED",
        }
    }
}
```

上記の属性はどちらも重要な診断情報を表し、非同期オペレーションの状態と、このビューで実行できるアクションを示します。これらの属性が取り得る値は次のとおりです。

1. Status.Action
   a. CREATE
   b. UPDATE
2. Status.State
   a. QUEUED
   b. IN_PROGRESS
   c. SUCCESS
   d. FAILED

重要な点として、データカタログビューの更新の中には、非同期オペレーションを必要としないものがあることにも注意してください。例えば、テーブルの Description 属性を更新しようとしているとします。これは非同期オペレーションを必要としないため、結果のテーブルメタデータには Status が含まれず、属性は NULL になります。

```
{
    "Table": {
        ...,
```

```
        "Description": "I changed this attribute!"
    }
}
```

次に、このトピックでは、上記のステータス情報が AWS Glue ビューで実行できるオペレーションにどのように影響するかについて説明します。

glue:CreateTable

この API は、Glue テーブルに対する `glue:CreateTable` の動作と比べて変更はありません。`CreateTable` は、まだ存在しない任意のテーブル名に対して呼び出すことができます。

glue:UpdateTable

このオペレーションは、次のステータス情報を持つ AWS Glue ビューでは実行できません。

1. Action == CREATE かつ State == QUEUED
2. Action == CREATE かつ State == IN_PROGRESS
3. Action == CREATE かつ State == FAILED
4. Action == UPDATE かつ State == QUEUED
5. Action == UPDATE かつ State == IN_PROGRESS

まとめると、データカタログビューは以下の要件を満たしている場合にのみ更新できます。

1. 最初に正常に作成された場合。
   a. Action == CREATE かつ State == SUCCESS
2. 非同期更新オペレーションの後、最終状態に達している場合。
   a. Action == UPDATE かつ State == SUCCESS
   b. Action == UPDATE かつ State == FAILED
3. 同期更新の結果として State 属性が `NULL` になっている場合。

glue:DeleteTable

このオペレーションは、 がどの AWS Glue テーブルに対してどのように`glue:DeleteTable`機能するかと比較して変更されません。データカタログビューは、その状態に関係なく削除できます。

glue:GetTable

このオペレーションは、 がどの AWS Glue テーブルに対してどのようにglue:GetTable機能するかと比較して変更されません。ただし、データカタログビューが最初に正常に作成されるまで (Action == CREATE and State == SUCCESS)、分析エンジンからデータカタログビューにクエリを実行することはできません。データカタログビューが最初に正常に作成された後は、ステータスに関係なくビューをクエリできます。

> **Note**
>
> このセクションのすべての情報は、GetTable、GetTables、SearchTables などのすべてのテーブル読み取り API に適用されます。

非同期オペレーションでの作成失敗シナリオの例

ここでは、ビューに対する CreateTable または UpdateTable API コールの結果として発生する可能性のある代表的なエラータイプの例を示します。SQL クエリの失敗のエラーサーフェスは大きいため、これらはすべてを網羅するものではありません。

シナリオ 1: Amazon Redshift クエリの失敗

Amazon Redshift に指定されたクエリにテーブル名のスペルミスが含まれているため、検証時にテーブルがデータカタログ内に見つかりません。結果のエラーは、ビューの GetTable レスポンスの Status フィールドに示されます。

GetTable リクエスト:

```
{
    "CatalogId": "123456789012",
    "DatabaseName": "async-view-test-db",
    "TableInput": {
        "Name": "view-athena-redshift-72",
        "Description": "This is an atomic operation",
        "StorageDescriptor": {
            "Columns": [
                { "Name": "col1", "Type": "int" },
                { "Name": "col2", "Type": "string" },
                { "Name": "col3", "Type": "double" }
            ]
        },
        "ViewDefinition": {
            "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
```

```
            "SubObjects": [ "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1" ],
            "Representations": [
                {
                    "Dialect": "ATHENA",
                    "DialectVersion": "3",
                    "ViewOriginalText": "SELECT * FROM \"gdc-view-playground-db\".
\"table_1\"",
                    "ValidationConnection": "athena-connection"
                },
                {
                    "Dialect": "REDSHIFT",
                    "DialectVersion": "1.0",
                    "ViewOriginalText": "SELECT * FROM \"gdc-view-playground-external-
schema\".\"table__1\";",
                    "ValidationConnection": "redshift-connection"
                }
            ]
        }
    }
}
```

GetTable レスポンス:

```
IncludeStatusDetails = FALSE
{
    "Table": {
        "Name": "view-athena-redshift-72",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:39:19-07:00",
        "UpdateTime": "2024-07-11T11:39:19-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
```

```
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:39:19-07:00",
            "UpdateTime": "2024-07-11T11:40:06-07:00",
            "Action": "CREATE",
            "State": "FAILED"
        }
    }
}

IncludeStatusDetails = TRUE
{
    "Table": {
        "Name": "view-athena-redshift-72",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:39:19-07:00",
        "UpdateTime": "2024-07-11T11:39:19-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:39:19-07:00",
            "UpdateTime": "2024-07-11T11:40:06-07:00",
            "Action": "CREATE",
            "State": "FAILED",
            "Error": {
                "ErrorCode": "QueryExecutionException",
                "ErrorMessage": "Error received during view SQL validation
 using a connection: [Connection Name: redshift-connection | Query Execution
 Id: ddb711d3-2415-4aa9-b251-6a76ab4f41b1 | Timestamp: Thu Jul 11 18:39:37 UTC
 2024]: Redshift returned error for the statement: ERROR: AwsClientException:
 EntityNotFoundException from glue - Entity Not Found"
```

```
            },
            "Details": {
                "RequestedChange": {
                    "Name": "view-athena-redshift-72",
                    "DatabaseName": "async-view-test-db",
                    "Description": "This is an atomic operation",
                    "Retention": 0,
                    "StorageDescriptor": {
                        "Columns": [
                            {
                                "Name": "col1",
                                "Type": "int"
                            },
                            {
                                "Name": "col2",
                                "Type": "string"
                            },
                            {
                                "Name": "col3",
                                "Type": "double"
                            }
                        ],
                        "Compressed": false,
                        "NumberOfBuckets": 0,
                        "SortColumns": [],
                        "StoredAsSubDirectories": false
                    },
                    "TableType": "VIRTUAL_VIEW",
                    "IsRegisteredWithLakeFormation": false,
                    "CatalogId": "123456789012",
                    "IsRowFilteringEnabled": false,
                    "VersionId": "-1",
                    "DatabaseId": "<databaseID>",
                    "ViewDefinition": {
                        "IsProtected": true,
                        "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
                        "SubObjects": [
                            "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1"
                        ],
                        "Representations": [
                            {
                                "Dialect": "ATHENA",
                                "DialectVersion": "3",
```

```
                                "ViewOriginalText": "SELECT * FROM \"gdc-view-
playground-db\".\"table_1\"",
                                "IsStale": false
                            },
                            {
                                "Dialect": "REDSHIFT",
                                "DialectVersion": "1.0",
                                "ViewOriginalText": "SELECT * FROM \"gdc-view-
playground-external-schema\".\"table__1\";",
                                "IsStale": false
                            }
                        ]
                    },
                    "IsMultiDialectView": true
                },
                "ViewValidations": [
                    {
                        "Dialect": "ATHENA",
                        "DialectVersion": "3",
                        "ViewValidationText": "SELECT * FROM \"gdc-view-playground-db
\".\"table_1\"",
                        "UpdateTime": "2024-07-11T11:40:06-07:00",
                        "State": "SUCCESS"
                    },
                    {
                        "Dialect": "REDSHIFT",
                        "DialectVersion": "1.0",
                        "ViewValidationText": "SELECT * FROM \"gdc-view-playground-
external-schema\".\"table__1\";",
                        "UpdateTime": "2024-07-11T11:39:37-07:00",
                        "State": "FAILED",
                        "Error": {
                            "ErrorCode": "QueryExecutionException",
                            "ErrorMessage": "Error received during view SQL validation
 using a connection: [Connection Name: redshift-connection | Query Execution Id:
 ddb711d3-2415-4aa9-b251-6a76ab4f41b1 | Timestamp: Thu
 Jul 11 18:39:37 UTC 2024]: Redshift returned error for the statement: ERROR:
 AwsClientException: EntityNotFoundException from glue - Entity Not Found"
                        }
                    }
                ]
            }
        }
    }
```

```
}
```

シナリオ 2: 無効な Amazon Redshift 接続

次の例の Amazon Redshift 接続は、指定されたクラスター/サーバーレスエンドポイントに存在しない Amazon Redshift データベースを参照しているため、正しくありません。Amazon Redshift はビューを検証できず、GetTable レスポンスの Status フィールドにはエラー (Amazon Redshift からの "State": "FAILED") が示されます。

GetTable リクエスト:

```
{
    "CatalogId": "123456789012",
    "DatabaseName": "async-view-test-db",
    "TableInput": {
        "Name": "view-athena-redshift-73",
        "Description": "This is an atomic operation",
        "StorageDescriptor": {
            "Columns": [
                { "Name": "col1", "Type": "int" },
                { "Name": "col2", "Type": "string" },
                { "Name": "col3", "Type": "double" }
            ]
        },
        "ViewDefinition": {
            "Definer": "arn:aws:iam::123456789012:role/GDCViewDdefiner",
            "SubObjects": [ "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1" ],
            "Representations": [
                {
                    "Dialect": "ATHENA",
                    "DialectVersion": "3",
                    "ViewOriginalText": "SELECT * FROM \"gdc-view-playground-db\".
\"table_1\"",
                    "ValidationConnection": "athena-connection"
                },
                {
                    "Dialect": "REDSHIFT",
                    "DialectVersion": "1.0",
                    "ViewOriginalText": "SELECT * FROM \"gdc-view-playground-external-
schema\".\"table_1\";",
                    "ValidationConnection": "redshift-connection-malformed"
                }
```

```
            ]
        }
    }
}
```

GetTable レスポンス:

```
IncludeStatusDetails = FALSE
{
    "Table": {
        "Name": "view-athena-redshift-73",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:43:27-07:00",
        "UpdateTime": "2024-07-11T11:43:27-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:43:27-07:00",
            "UpdateTime": "2024-07-11T11:43:40-07:00",
            "Action": "CREATE",
            "State": "FAILED"
        }
    }
}

IncludeStatusDetails = TRUE
{
    "Table": {
        "Name": "view-athena-redshift-73",
        "DatabaseName": "async-view-test-db",
        "Description": "",
```

```
        "CreateTime": "2024-07-11T11:43:27-07:00",
        "UpdateTime": "2024-07-11T11:43:27-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:43:27-07:00",
            "UpdateTime": "2024-07-11T11:43:40-07:00",
            "Action": "CREATE",
            "State": "FAILED",
            "Error": {
                "ErrorCode": "QueryExecutionException",
                "ErrorMessage": "Error received during view SQL validation using a
connection: [Connection Name: redshift-connection-malformed | Query Execution Id:
69bfafd4-3d51-4cb0-9320-7ce5404b1809 | Timestamp: Thu Jul 11 18:43:38 UTC 2024]:
Redshift returned error for the statement: FATAL: database \"devooo\" does not exist"
            },
            "Details": {
                "RequestedChange": {
                    "Name": "view-athena-redshift-73",
                    "DatabaseName": "async-view-test-db",
                    "Description": "This is an atomic operation",
                    "Retention": 0,
                    "StorageDescriptor": {
                        "Columns": [
                            {
                                "Name": "col1",
                                "Type": "int"
                            },
                            {
                                "Name": "col2",
                                "Type": "string"
                            },
                            {
```

```
                                "Name": "col3",
                                "Type": "double"
                            }
                        ],
                        "Compressed": false,
                        "NumberOfBuckets": 0,
                        "SortColumns": [],
                        "StoredAsSubDirectories": false
                    },
                    "TableType": "VIRTUAL_VIEW",
                    "IsRegisteredWithLakeFormation": false,
                    "CatalogId": "123456789012",
                    "IsRowFilteringEnabled": false,
                    "VersionId": "-1",
                    "DatabaseId": "<databaseID>",
                    "ViewDefinition": {
                        "IsProtected": true,
                        "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
                        "SubObjects": [
                            "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1"
                        ],
                        "Representations": [
                            {
                                "Dialect": "ATHENA",
                                "DialectVersion": "3",
                                "ViewOriginalText": "SELECT * FROM \"gdc-view-
playground-db\".\"table_1\"",
                                "IsStale": false
                            },
                            {
                                "Dialect": "REDSHIFT",
                                "DialectVersion": "1.0",
                                "ViewOriginalText": "SELECT * FROM \"gdc-view-
playground-external-schema\".\"table_1\";",
                                "IsStale": false
                            }
                        ]
                    },
                    "IsMultiDialectView": true
                },
                "ViewValidations": [
                    {
                        "Dialect": "ATHENA",
```

```
                        "DialectVersion": "3",
                        "ViewValidationText": "SELECT * FROM \"gdc-view-playground-db
\".\"table_1\"",
                        "UpdateTime": "2024-07-11T11:43:40-07:00",
                        "State": "SUCCESS"
                    },
                    {
                        "Dialect": "REDSHIFT",
                        "DialectVersion": "1.0",
                        "ViewValidationText": "SELECT * FROM \"gdc-view-playground-
external-schema\".\"table_1\";",
                        "UpdateTime": "2024-07-11T11:43:38-07:00",
                        "State": "FAILED",
                        "Error": {
                            "ErrorCode": "QueryExecutionException",
                            "ErrorMessage": "Error received during view SQL validation
 using a connection: [Connection Name: redshift-connection-malformed | Query Execution
 Id: 69bfafd4-3d51-4cb0-9320-7ce5404b1809 | Time
stamp: Thu Jul 11 18:43:38 UTC 2024]: Redshift returned error for the statement: FATAL:
 database \"devooo\" does not exist"
                        }
                    }
                ]
            }
        }
    }
}
```

シナリオ 3: Athena クエリの失敗

ここでの Athena の SQL は、クエリにデータベース名のスペルミスが含まれているため無効です。Athena のクエリ検証によってこれが検出され、結果のエラーが GetTable 呼び出しの Status オブジェクトを通じて表面化されます。

GetTable リクエスト:

```
{
    "CatalogId": "123456789012",
    "DatabaseName": "async-view-test-db",
    "TableInput": {
        "Name": "view-athena-redshift-70",
```

```
        "Description": "This is an atomic operation",
        "StorageDescriptor": {
            "Columns": [
                { "Name": "col1", "Type": "int" },
                { "Name": "col2", "Type": "string" },
                { "Name": "col3", "Type": "double" }
            ]
        },
        "ViewDefinition": {
            "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
            "SubObjects": [ "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1" ],
            "Representations": [
                {
                    "Dialect": "ATHENA",
                    "DialectVersion": "3",
                    "ViewOriginalText": "SELECT * FROM \"gdc--view-playground-db\".
\"table_1\"",
                    "ValidationConnection": "athena-connection"
                },
                {
                    "Dialect": "REDSHIFT",
                    "DialectVersion": "1.0",
                    "ViewOriginalText": "SELECT * FROM \"gdc-view-playground-external-
schema\".\"table_1\";",
                    "ValidationConnection": "redshift-connection"
                }
            ]
        }
    }
}
```

GetTable レスポンス:

```
IncludeStatusDetails = FALSE
{
    "Table": {
        "Name": "view-athena-redshift-70",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:09:53-07:00",
        "UpdateTime": "2024-07-11T11:09:53-07:00",
        "Retention": 0,
```

```
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:09:54-07:00",
            "UpdateTime": "2024-07-11T11:10:41-07:00",
            "Action": "CREATE",
            "State": "FAILED",
        }
    }
}

IncludeStatusDetails = TRUE
{
    "Table": {
        "Name": "view-athena-redshift-70",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:09:53-07:00",
        "UpdateTime": "2024-07-11T11:09:53-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:09:54-07:00",
```

```
            "UpdateTime": "2024-07-11T11:10:41-07:00",
            "Action": "CREATE",
            "State": "FAILED",
            "Error": {
                "ErrorCode": "QueryExecutionException",
                "ErrorMessage": "Error received during view SQL validation using
 a connection: [Connection Name: athena-connection | Query Execution Id: d9bb1e6d-
ce26-4b35-8276-8a199af966aa | Timestamp: Thu Jul 11 18:10:
41 UTC 2024]: Athena validation FAILED: {ErrorCategory: 2,ErrorType: 1301,Retryable:
 false,ErrorMessage: line 1:118: Schema 'gdc--view-playground-db' does not exist}"
            },
            "Details": {
                "RequestedChange": {
                    "Name": "view-athena-redshift-70",
                    "DatabaseName": "async-view-test-db",
                    "Description": "This is an atomic operation",
                    "Retention": 0,
                    "StorageDescriptor": {
                        "Columns": [
                            {
                                "Name": "col1",
                                "Type": "int"
                            },
                            {
                                "Name": "col2",
                                "Type": "string"
                            },
                            {
                                "Name": "col3",
                                "Type": "double"
                            }
                        ],
                        "Compressed": false,
                        "NumberOfBuckets": 0,
                        "SortColumns": [],
                        "StoredAsSubDirectories": false
                    },
                    "TableType": "VIRTUAL_VIEW",
                    "IsRegisteredWithLakeFormation": false,
                    "CatalogId": "123456789012",
                    "IsRowFilteringEnabled": false,
                    "VersionId": "-1",
                    "DatabaseId": "<databaseID>",
                    "ViewDefinition": {
```

```
                        "IsProtected": true,
                        "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
                        "SubObjects": [
                            "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1"
                        ],
                        "Representations": [
                            {
                                "Dialect": "ATHENA",
                                "DialectVersion": "3",
                                "ViewOriginalText": "SELECT * FROM \"gdc--view-
playground-db\".\"table_1\"",
                                "IsStale": false
                            },
                            {
                                "Dialect": "REDSHIFT",
                                "DialectVersion": "1.0",
                                "ViewOriginalText": "SELECT * FROM \"gdc-view-
playground-external-schema\".\"table_1\";",
                                "IsStale": false
                            }
                        ]
                    },
                    "IsMultiDialectView": true
                },
                "ViewValidations": [
                    {
                        "Dialect": "ATHENA",
                        "DialectVersion": "3",
                        "ViewValidationText": "SELECT * FROM \"gdc--view-playground-db
\".\"table_1\"",
                        "UpdateTime": "2024-07-11T11:10:41-07:00",
                        "State": "FAILED",
                        "Error": {
                            "ErrorCode": "QueryExecutionException",
                            "ErrorMessage": "Error received during view SQL validation
 using a connection: [Connection Name: athena-connection | Query Execution Id:
 d9bb1e6d-ce26-4b35-8276-8a199af966aa | Timestamp: Thu J
ul 11 18:10:41 UTC 2024]: Athena validation FAILED: {ErrorCategory: 2,ErrorType:
 1301,Retryable: false,ErrorMessage: line 1:118: Schema 'gdc--view-playground-db' does
 not exist}"
                        }
                    },
                    {
```

```
                        "Dialect": "REDSHIFT",
                        "DialectVersion": "1.0",
                        "ViewValidationText": "SELECT * FROM \"gdc-view-playground-
external-schema\".\"table_1\";",
                        "UpdateTime": "2024-07-11T11:10:41-07:00",
                        "State": "SUCCESS"
                    }
                ]
            }
        }
    }
}
```

シナリオ 4: ストレージ記述子の不一致

Athena ダイアレクトに指定された SQL では col1 と col2 が選択されますが、Redshift の SQL では col1 のみが選択されます。これにより、ストレージ記述子の不一致エラーが発生します。

GetTable リクエスト:

```
{
    "CatalogId": "123456789012",
    "DatabaseName": "async-view-test-db",
    "TableInput": {
        "Name": "view-athena-redshift-71",
        "Description": "This is an atomic operation",
        "StorageDescriptor": {
            "Columns": [
                { "Name": "col1", "Type": "int" },
                { "Name": "col2", "Type": "string" },
                { "Name": "col3", "Type": "double" }
            ]
        },
        "ViewDefinition": {
            "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
            "SubObjects": [ "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1" ],
            "Representations": [
                {
                    "Dialect": "ATHENA",
                    "DialectVersion": "3",
                    "ViewOriginalText": "SELECT col1, col2 FROM \"gdc-view-playground-
db\".\"table_1\"",
```

```
                    "ValidationConnection": "athena-connection"
                },
                {
                    "Dialect": "REDSHIFT",
                    "DialectVersion": "1.0",
                    "ViewOriginalText": "SELECT col1 FROM \"gdc-view-playground-
external-schema\".\"table_1\";",
                    "ValidationConnection": "redshift-connection"
                }
            ]
        }
    }
}
```

GetTable レスポンス:

```
IncludeStatusDetails = FALSE

{
    "Table": {
        "Name": "view-athena-redshift-71",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:22:02-07:00",
        "UpdateTime": "2024-07-11T11:22:02-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:22:02-07:00",
            "UpdateTime": "2024-07-11T11:23:19-07:00",
            "Action": "CREATE",
            "State": "FAILED"
```

```
        }
    }
}

IncludeStatusDetails = TRUE

{
    "Table": {
        "Name": "view-athena-redshift-71",
        "DatabaseName": "async-view-test-db",
        "Description": "",
        "CreateTime": "2024-07-11T11:22:02-07:00",
        "UpdateTime": "2024-07-11T11:22:02-07:00",
        "Retention": 0,
        "ViewOriginalText": "",
        "ViewExpandedText": "",
        "TableType": "",
        "CreatedBy": "arn:aws:iam::123456789012:user/zcaisse",
        "IsRegisteredWithLakeFormation": false,
        "CatalogId": "123456789012",
        "IsRowFilteringEnabled": false,
        "VersionId": "-1",
        "DatabaseId": "<databaseID>",
        "IsMultiDialectView": false,
        "Status": {
            "RequestedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "UpdatedBy": "arn:aws:iam::123456789012:user/zcaisse",
            "RequestTime": "2024-07-11T11:22:02-07:00",
            "UpdateTime": "2024-07-11T11:23:19-07:00",
            "Action": "CREATE",
            "State": "FAILED",
            "Error": {
                "ErrorCode": "InvalidInputException",
                "ErrorMessage": "Engine and existing storage descriptor mismatch"
            },
            "Details": {
                "RequestedChange": {
                    "Name": "view-athena-redshift-71",
                    "DatabaseName": "async-view-test-db",
                    "Description": "This is an atomic operation",
                    "Retention": 0,
                    "StorageDescriptor": {
                        "Columns": [
                            {
```

```
                                "Name": "col1",
                                "Type": "int"
                            },
                            {
                                "Name": "col2",
                                "Type": "string"
                            },
                            {
                                "Name": "col3",
                                "Type": "double"
                            }
                        ],
                        "Compressed": false,
                        "NumberOfBuckets": 0,
                        "SortColumns": [],
                        "StoredAsSubDirectories": false
                    },
                    "TableType": "VIRTUAL_VIEW",
                    "IsRegisteredWithLakeFormation": false,
                    "CatalogId": "123456789012",
                    "IsRowFilteringEnabled": false,
                    "VersionId": "-1",
                    "DatabaseId": "<databaseID>",
                    "ViewDefinition": {
                        "IsProtected": true,
                        "Definer": "arn:aws:iam::123456789012:role/GDCViewDefiner",
                        "SubObjects": [
                            "arn:aws:glue:us-east-1:123456789012:table/gdc-view-
playground-db/table_1"
                        ],
                        "Representations": [
                            {
                                "Dialect": "ATHENA",
                                "DialectVersion": "3",
                                "ViewOriginalText": "SELECT col1, col2 FROM \"gdc-view-
playground-db\".\"table_1\"",
                                "IsStale": false
                            },
                            {
                                "Dialect": "REDSHIFT",
                                "DialectVersion": "1.0",
                                "ViewOriginalText": "SELECT col1 FROM \"gdc-view-
playground-external-schema\".\"table_1\";",
                                "IsStale": false
```

```
                            }
                        ]
                    },
                    "IsMultiDialectView": true
                },
                "ViewValidations": [
                    {
                        "Dialect": "ATHENA",
                        "DialectVersion": "3",
                        "ViewValidationText": "SELECT col1, col2 FROM \"gdc-view-
playground-db\".\"table_1\"",
                        "UpdateTime": "2024-07-11T11:23:19-07:00",
                        "State": "FAILED",
                        "Error": {
                            "ErrorCode": "InvalidInputException",
                            "ErrorMessage": "Engine and existing storage descriptor
 mismatch"
                        }
                    },
                    {
                        "Dialect": "REDSHIFT",
                        "DialectVersion": "1.0",
                        "ViewValidationText": "SELECT col1 FROM \"gdc-view-playground-
external-schema\".\"table_1\";",
                        "UpdateTime": "2024-07-11T11:22:49-07:00",
                        "State": "FAILED",
                        "Error": {
                            "ErrorCode": "InvalidInputException",
                            "ErrorMessage": "Engine and existing storage descriptor
 mismatch"
                        }
                    }
                ]
            }
        }
    }
}
```

データカタログビューに対する許可の付与

でビューを作成したら AWS Glue Data Catalog、ビューに対するデータレイクのアクセス許可を、、組織 AWS アカウント、組織単位全体のプリンシパルに付与できます。アクセス許可は、LF タグまたは名前付きリソース方式を使用して付与できます。リソースのタグ付けの詳細については、

「Lake Formation のタグベースのアクセス制御」を参照してください。ビューに対するアクセス許可の直接付与の詳細については、「名前付きリソース方式を使用したビューに対するアクセス権限の付与」を参照してください。

マテリアライズドビュー

トピック

- マテリアライズドビューを他のビュータイプと区別する
- ユースケース
- 主要なコンセプト
- マテリアライズドビューのアクセス許可
- マテリアライズドビューの作成と管理
- ストレージとデータアクセス
- アクセス AWS Lake Formation 許可との統合
- モニタリングとデバッグ
- 更新ジョブの管理
- モニタリングとトラブルシューティング
- 考慮事項と制限事項

AWS Glue データカタログでは、マテリアライズドビューは、SQL クエリの事前計算された結果を Apache Iceberg 形式で保存するマネージドテーブルです。アクセスされるたびにクエリを実行する標準の Data Catalog ビューとは異なり、マテリアライズドビューはクエリ結果を物理的に保存し、基盤となるソーステーブルが変更されるたびに更新します。Amazon Athena、Amazon EMR、または で Apache Spark バージョン 3.5.6 以降を使用してマテリアライズドビューを作成できます AWS Glue。

マテリアライズドビューは、 AWS Glue Data Catalog に登録された Apache Iceberg テーブルを参照し、事前計算されたデータは Amazon S3 Tables バケットまたは Amazon S3 汎用バケットに Apache Iceberg テーブルとして保存されるため、Amazon Athena、Amazon Redshift、サードパーティーの Iceberg 互換エンジンなどの複数のクエリエンジンからアクセスできます。

マテリアライズドビューを他のビュータイプと区別する

マテリアライズドビューは、 AWS Glue Data Catalog ビュー、Apache Spark ビュー、Amazon Athena ビューと根本的に異なります。データカタログビューは、アクセスされるたびに SQL クエリ定義を実行する仮想テーブルですが、マテリアライズドビューは事前に計算されたクエリ結果を物理

的に保存します。これにより、冗長な計算が不要になり、頻繁にアクセスされる複雑な変換のクエリパフォーマンスが大幅に向上します。

マテリアライズドビューは、ETL ジョブまたはカスタム Spark AWS Glue ジョブで構築された従来のデータ変換パイプラインとも異なります。変更検出、増分更新、ワークフローオーケストレーションを処理するカスタムコードを記述する代わりに、標準の SQL 構文を使用してマテリアライズドビューを定義します。Data Catalog AWS Glue は、フルマネージド型のコンピューティングインフラストラクチャを使用して、ソーステーブルを自動的にモニタリングし、変更を検出し、マテリアライズドビューを更新します。

ユースケース

マテリアライズドビューの重要なユースケースは次のとおりです。

- 複雑な分析クエリの高速化 – 高価な結合、集計、ウィンドウ関数を事前に計算するマテリアライズドビューを作成します。Spark エンジンは、事前計算された結果を使用するように後続のクエリを自動的に書き換え、クエリのレイテンシーとコンピューティングコストを削減します。
- データ変換パイプラインの簡素化 – 変更検出、増分更新、ワークフローオーケストレーションを処理する複雑な ETL ジョブを、シンプルな SQL ベースのマテリアライズドビュー定義に置き換えます。 AWS Glue Data Catalog は、運用上の複雑さをすべて自動的に管理します。
- 管理されたデータアクセスでセルフサービス分析を有効にする – raw データをビジネス対応のデータセットに変換する厳選されたマテリアライズドビューを作成します。基盤となるソーステーブルを公開することなくマテリアライズドビューへのアクセスをユーザーに付与し、セルフサービス分析を強化しながらセキュリティ管理を簡素化します。
- 機械学習の特徴量エンジニアリングを最適化する – ML モデルの特徴量変換を実装するマテリアライズドビューを定義します。自動更新機能により、ソースデータの進化に合わせて機能ストアを最新の状態に保ち、増分更新によりコンピューティングコストを最小限に抑えます。
- 効率的なデータ共有の実装 – 特定のコンシューマーのデータをフィルタリングおよび変換するマテリアライズドビューを作成します。を使用してアカウントとリージョン間でマテリアライズドビューを共有するため AWS Lake Formation、一元化されたガバナンスを維持しながら、データの重複を排除できます。

主要なコンセプト

自動更新

自動更新は、ソーステーブルを継続的にモニタリングし、定義したスケジュールに従ってマテリアライズドビューを更新する機能です。マテリアライズドビューを作成するときは、1 時間間隔の時

間ベースのスケジューリングを使用して更新頻度を指定できます。 AWS Glue Data Catalog は、マネージド Spark コンピューティングインフラストラクチャを使用してバックグラウンドで更新オペレーションを実行し、変更検出と増分更新のすべての側面を透過的に処理します。

更新間隔の間にソースデータが変更されると、マテリアライズドビューは一時的に古くなります。マテリアライズドビューに直接アクセスするクエリは、次のスケジュールされた更新が完了するまで、古い結果が返されることがあります。最新のデータへの即時アクセスが必要なシナリオでは、`REFRESH MATERIALIZED VIEW`SQL コマンドを使用して手動更新を実行できます。

増分更新

増分更新は、マテリアライズドビュー全体を再計算するのではなく、前回の更新以降にソーステーブルで変更されたデータのみを処理する最適化手法です。 AWS Glue Data Catalog は Apache Iceberg のメタデータレイヤーを活用して、ソーステーブルの変更を効率的に追跡し、マテリアライズドビューのどの部分に更新が必要かを判断します。

このアプローチは、フル更新オペレーションと比較して、コンピューティングコストと更新期間を大幅に削減します。特に、更新サイクル間でデータ変更の割合がごくわずかである大規模なデータセットの場合です。増分更新メカニズムは自動的に動作します。変更されたデータを検出または処理するためにカスタムロジックを記述する必要はありません。

自動クエリ書き換え

自動クエリ書き換えは、Amazon Athena、Amazon EMR、および の Spark エンジンで使用できるクエリ最適化機能です AWS Glue。ベーステーブルに対してクエリを実行すると、Spark オプティマイザはクエリプランを分析し、使用可能なマテリアライズドビューがクエリをより効率的に満たすことができるかどうかを自動的に判断します。適切なマテリアライズドビューが存在する場合、オプティマイザはベーステーブルを処理する代わりに、事前に計算された結果を使用するようにクエリを透過的に書き換えます。

この最適化は、アプリケーションコードやクエリステートメントを変更することなく行われます。Spark オプティマイザは、自動クエリ書き換えがマテリアライズドビューが最新で、正確な結果を生成できる場合にのみ適用されるようにします。マテリアライズドビューが古くなったり、クエリ要件に完全に一致しない場合、オプティマイザはベーステーブルに対して元のクエリプランを実行し、パフォーマンスよりも正確性を優先します。

ビュー定義ロール

マテリアライズドビューは、ビュー定義ロールと呼ばれる、マテリアライズドビューを作成した IAM ロールのアクセス許可に基づいて動作します。定義者ロールには、マテリアライズドビュー定義で参照されるすべてのベーステーブルへの読み取りアクセスと、ターゲットデータベースへのテー

ブルアクセス許可の作成が必要です。Data Catalog AWS Glue がマテリアライズドビューを更新すると、ソーステーブルにアクセスして更新された結果を書き込むための定義ロールを引き受けます。

このセキュリティモデルを使用すると、基になるソーステーブルに対する直接アクセス許可を付与することなく、マテリアライズドビューへのアクセス権をユーザーに付与できます。ビュー定義ロールがベーステーブルにアクセスできなくなった場合、アクセス許可が復元されるまで後続の更新オペレーションは失敗します。

マテリアライズドビューのアクセス許可

マテリアライズドビューを作成および管理するには、 AWS Lake Formation アクセス許可を設定する必要があります。マテリアライズドビューを作成する IAM ロール (定義ロール) には、ソーステーブルとターゲットデータベースに対する特定のアクセス許可が必要です。

定義者ロールに必要なアクセス許可

定義者ロールには、次の Lake Formation アクセス許可が必要です。

- ソーステーブル – 行、列、またはセルフィルターが適用されていない SELECT または ALL アクセス許可
- ターゲットデータベース – CREATE_TABLE アクセス許可
- AWS Glue データカタログ – GetTable および CreateTable API アクセス許可

マテリアライズドビューを作成すると、定義ロールの ARN がビュー定義に保存されます。 AWS Glue データカタログは、自動更新オペレーションを実行するときにこのロールを引き受けます。定義者ロールがソーステーブルへのアクセスを失った場合、アクセス許可が復元されるまで更新オペレーションは失敗します。

AWS Glue ジョブの IAM アクセス許可

AWS Glue ジョブの IAM ロールには、次のアクセス許可が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetCatalog",
                "glue:GetCatalogs",
                "glue:GetTable",
```

```
                "glue:GetTables",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*:/aws-glue/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "lakeformation:GetDataAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

マテリアライズドビューの自動更新に使用するロールには、ロールに対する iam:PassRole アクセス許可が必要です。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/materialized-view-role-name"
      ]
    }
  ]
}
```

Glue でマテリアライズドビューを自動的に更新するには、サービスがロールを引き受けることができる次の信頼ポリシーもロールに必要です。

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

```
}
```

マテリアライズドビューが S3 Tables バケットに保存されている場合は、ロールに次のアクセス許可も追加する必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3tables:PutTableMaintenanceConfiguration"
      ],
      "Resource": "arn:aws:s3tables:*:123456789012:*"
    }
  ]
}
```

マテリアライズドビューへのアクセスを付与する

マテリアライズドビューをクエリするアクセス許可を他のユーザーに付与するには、 AWS Lake Formation を使用してマテリアライズドビューテーブルに対する SELECT アクセス許可を付与します。ユーザーは、マテリアライズドビューをクエリできます。基盤となるソーステーブルに直接アクセスする必要はありません。

Lake Formation アクセス許可の設定の詳細については、「 AWS Lake Formation デベロッパーガイド」の「データカタログリソースに対するアクセス許可の付与と取り消し」を参照してください。

マテリアライズドビューの作成と管理

Spark エンジンの `CREATE MATERIALIZED VIEW` SQL ステートメントを使用してマテリアライズドビューを作成します。ビュー定義は、変換ロジック、ターゲットデータベースとテーブル名、およびオプションの更新設定を定義する SQL クエリを指定します。集計、複数のテーブル間の結合、フィルター、ウィンドウ関数など、複雑な変換を定義できます。

```
CREATE MATERIALIZED VIEW sales_summary
AS
SELECT
```

```
    region,
    product_category,
    SUM(sales_amount) as total_sales,
    COUNT(DISTINCT customer_id) as unique_customers
FROM sales_transactions
WHERE transaction_date >= current_date - interval '90' day
GROUP BY region, product_category;
```

自動更新を設定するには、ビュー定義に更新スケジュールを含めます。

```
CREATE MATERIALIZED VIEW sales_summary
SCHEDULE REFRESH EVERY 1 HOUR
AS
SELECT region, product_category, SUM(sales_amount) as total_sales
FROM sales_transactions
GROUP BY region, product_category;
```

マテリアライズドビューは、 REFRESH MATERIALIZED VIEW コマンドを使用していつでも手動で更新できます。

```
REFRESH MATERIALIZED VIEW sales_summary;
```

既存のマテリアライズドビューの更新スケジュールを変更するには、 ALTER MATERIALIZED VIEWステートメントを使用します。

```
ALTER MATERIALIZED VIEW sales_summary
ADD SCHEDULE REFRESH EVERY 2 HOURS;
```

ネストされたマテリアライズドビュー

他のマテリアライズドビューをベーステーブルとして参照するマテリアライズドビューを作成し、複数ステージのデータ変換を有効にすることができます。ネストされたマテリアライズドビューを作成すると、 AWS Glue データカタログは依存関係を追跡し、マテリアライズドビュー階層を通じて更新を自動的に伝達します。ベースマテリアライズドビューが更新されると、それに依存するすべてのダウンストリームマテリアライズドビューがそれに応じて更新されます。

この機能を使用すると、複雑な変換を論理ステージに分解できるため、保守性が向上し、データの鮮度要件に基づいて変換レイヤーを選択的に更新できます。

ストレージとデータアクセス

マテリアライズドビューは、事前計算された結果を Apache Iceberg テーブルとして S3 Tables バケットまたは AWS アカウント内の汎用 S3 バケットに保存します。 AWS Glue Data Catalog は、S3 Tables の自動最適化機能を通じて、圧縮やスナップショットの保持など、Iceberg テーブルのメンテナンスのあらゆる側面を管理します。

マテリアライズドビューは Iceberg テーブルとして保存されるため、Amazon Athena、Amazon Redshift、サードパーティーの分析プラットフォームなど、Iceberg 互換エンジンから直接読み取ることができます。このマルチエンジンアクセシビリティにより、事前計算されたデータは、データ重複や形式変換なしで分析エコシステム全体でアクセス可能になります。

アクセス AWS Lake Formation 許可との統合

を使用して AWS Lake Formation 、マテリアライズドビューに対するきめ細かなアクセス許可を管理できます。ビュー作成者は自動的にマテリアライズドビューの所有者になり、 AWS Lake Formationの名前付きリソースメソッドまたは LF タグを使用して他のユーザーまたはロールにアクセス許可を付与できます。

マテリアライズドビューに対するSELECTアクセス許可をユーザーに付与すると、ユーザーは基盤となるソーステーブルにアクセスすることなく、事前に計算された結果をクエリできます。このセキュリティモデルは、データアクセス管理を簡素化し、最小特権の原則を実装して、ユーザーが必要とする特定のデータ変換にのみアクセスできるようにします。

AWS Lake Formationのクロスアカウント共有機能を使用して、 AWS アカウント、 AWS 組織、組織単位間でマテリアライズドビューを共有できます。リソースリンクを使用して AWS リージョン間でマテリアライズドビューにアクセスすることもできます。これにより、分散データアクセスによる一元化されたデータガバナンスが可能になります。

モニタリングとデバッグ

AWS Glue Data Catalog は、すべてのマテリアライズドビューの更新オペレーションと関連するメトリクスを Amazon CloudWatch に発行します。CloudWatch メトリクスを使用して、更新の開始時刻、終了時刻、期間、処理されたデータボリューム、更新ステータスをモニタリングできます。更新オペレーションが失敗すると、エラーメッセージと診断情報が CloudWatch Logs にキャプチャされます。

更新ジョブが予想期間を超えたとき、または繰り返し失敗したときに通知を受信するように CloudWatch アラームを設定できます。 AWS Glue データカタログは、更新実行の成功と失敗の両

方について変更イベントを に発行するため、マテリアライズドビューオペレーションをより広範なワークフローオートメーションに統合できます。

マテリアライズドビューの現在のステータスを確認するには、DESCRIBE MATERIALIZED VIEWSQL コマンドを使用します。このコマンドは、古いステータス、最終更新タイムスタンプ、更新スケジュール設定などのメタデータを返します。

更新ジョブの管理

手動更新の開始

スケジュールされた間隔外に即時更新をトリガーします。

必要なアクセス許可: API コールを行うために使用される AWS 認証情報には、マテリアライズドビューに対するglue:GetTableアクセス許可が必要です。

S3 Tables Catalog の場合:

```
aws glue start-materialized-view-refresh-task-run \
    --catalog-id <ACCOUNT_ID>:s3tablescatalog/<CATALOG_NAME> \
    --database-name <DATABASE_NAME> \
    --table-name <MV_TABLE_NAME>
```

ルートカタログの場合:

```
aws glue start-materialized-view-refresh-task-run \
    --catalog-id <ACCOUNT_ID> \
    --database-name <DATABASE_NAME> \
    --table-name <MV_TABLE_NAME>
```

更新ステータスの確認

特定の更新ジョブのステータスを取得します。

```
aws glue get-materialized-view-refresh-task-run \
    --catalog-id <CATALOG_ID> \
    --materialized-view-refresh-task-run-id <TASK_RUN_ID>
```

更新履歴の一覧表示

マテリアライズドビューのすべての更新ジョブを表示します。

```
aws glue list-materialized-view-refresh-task-runs \
    --catalog-id <CATALOG_ID> \
    --database-name <DATABASE_NAME> \
    --table-name <MV_TABLE_NAME>
```

> **Note**
>
> S3 テーブル<ACCOUNT_ID>:s3tablescatalog/<CATALOG_NAME>の場合は 、ルートカタログ<ACCOUNT_ID>の場合は を使用します。

実行中の更新の停止

進行中の更新ジョブをキャンセルします。

```
aws glue stop-materialized-view-refresh-task-run \
    --catalog-id <CATALOG_ID> \
    --database-name <DATABASE_NAME> \
    --table-name <MV_TABLE_NAME>
```

モニタリングとトラブルシューティング

マテリアライズドビューの更新ジョブをモニタリングするには、次の 3 つの方法があります。

CloudWatch メトリクス

CloudWatch のすべてのマテリアライズドビュー更新ジョブの集計メトリクスを表示します。

使用可能なメトリクス:

- AWSディメンションを含む /Glue 名前空間:
  - CatalogId: カタログ識別子
  - DatabaseName: マテリアライズドビューを含むデータベース
  - TableName: マテリアライズドビュー名
  - TaskType: をMaterializedViewRefresh」に設定する

コンソールでの表示:

1. CloudWatch コンソール → メトリクスに移動する

2. Select AWS/Glue 名前空間

3. ディメンションでフィルタリング: CatalogId、DatabaseName、TableName、TaskType

4. ジョブの成功、失敗、期間に関するメトリクスを表示する

CloudWatch メトリクスクエリの例:

```
{AWS/Glue,CatalogId,DatabaseName,TableName,TaskType} MaterializedViewRefresh
```

の使用 AWS CLI:

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/Glue \
    --metric-name <MetricName> \
    --dimensions Name=CatalogId,Value=<CATALOG_ID> \
                 Name=DatabaseName,Value=<DATABASE_NAME> \
                 Name=TableName,Value=<TABLE_NAME> \
                 Name=TaskType,Value=MaterializedViewRefresh \
    --start-time <START_TIME> \
    --end-time <END_TIME> \
    --period 3600 \
    --statistics Sum \
    --region <REGION>
```

CloudWatch Logs

個々の更新タスク実行の詳細な実行ログを表示します。

ロググループ: `/aws-glue/materialized-views/<task_run_id>`

`<task_run_id>` は UUID (abc12345-def6-7890-ghij-klmnopqrstuv など) です。

ログの表示:

```
# List log streams for a task run
aws logs describe-log-streams \
    --log-group-name /aws-glue/materialized-views/<TASK_RUN_ID> \
    --region <REGION>

# Get log events
aws logs get-log-events \
```

```
    --log-group-name /aws-glue/materialized-views/<TASK_RUN_ID> \
    --log-stream-name <LOG_STREAM_NAME> \
    --region <REGION>
```

CloudWatch コンソールの場合:

1. CloudWatch → ロググループに移動する
2. /aws-glue/materialized-views/ を検索する
3. タスク実行 ID を持つロググループを選択する
4. 詳細な実行ログ、エラー、Spark ジョブ出力を表示する

通知

更新ジョブの状態の変更に関するリアルタイム通知のイベントをサブスクライブします。

使用可能なイベントタイプ:

- Glue マテリアライズドビューの更新タスクが開始されました
- Glue マテリアライズドビューの更新タスクが正常に完了しました
- Glue マテリアライズドビューの更新タスクに失敗しました
- Glue マテリアライズドビューの自動更新呼び出しの失敗

ルールの作成:

```
aws events put-rule \
    --name materialized-view-refresh-notifications \
    --event-pattern '{
        "source": ["aws.glue"],
        "detail-type": [
            "Glue Materialized View Refresh Task Started",
            "Glue Materialized View Refresh Task Succeeded",
            "Glue Materialized View Refresh Task Failed",
            "Glue Materialized View Auto-Refresh Invocation Failure"
        ]
    }' \
    --region <REGION>
```

ターゲットの追加 (SNS トピックなど)

```
aws events put-targets \
    --rule materialized-view-refresh-notifications \
    --targets "Id"="1","Arn"="arn:aws:sns:<REGION>:<ACCOUNT_ID>:<TOPIC_NAME>" \
    --region <REGION>
```

更新ステータスの表示

AWS Glue API を使用してマテリアライズドビューの更新ジョブのステータスを確認します。

```
aws glue get-materialized-view-refresh-task-run \
    --catalog-id <CATALOG_ID> \
    --materialized-view-refresh-task-run-id <TASK_RUN_ID> \
    --region <REGION>
```

または、最近のすべての更新実行を一覧表示します。

```
aws glue list-materialized-view-refresh-task-runs \
    --catalog-id <CATALOG_ID> \
    --database-name <DATABASE_NAME> \
    --table-name <MV_TABLE_NAME> \
    --region <REGION>
```

これは以下を示しています。

- 最終更新時間
- 更新ステータス (成功、失敗、実行中、停止)
- タスク実行 ID
- エラーメッセージ (失敗した場合)

一般的な更新状態:

- RUNNING: 更新ジョブは現在実行中です
- 成功: 更新が正常に完了しました
- 失敗: 更新でエラーが発生しました
- STOPPED: 更新が手動でキャンセルされました

更新失敗のトラブルシューティング:

更新が失敗した場合は、以下を確認してください。

1. IAM アクセス許可: 定義者ロールがすべてのベーステーブルとマテリアライズドビューの場所にアクセスできることを確認します。
2. ベーステーブルの可用性: 参照されるすべてのテーブルが存在し、アクセス可能であることを確認する
3. クエリの有効性: SQL クエリが Spark SQL ダイアレクトに対して有効であることを確認する
4. リソース制限: アカウントの同時更新制限に達したかどうかを確認します。

GetMaterializedViewRefreshTaskRun API を使用して、詳細なエラーメッセージを取得します。

考慮事項と制限事項

- マテリアライズドビューは、 AWS Glue データカタログに登録された Apache Iceberg テーブルをベーステーブルとしてのみ参照できます。
- ビューの作成と自動クエリの書き換えは、Amazon Athena、Amazon EMR、および AWS Glue (バージョン 5.1) 全体で Apache Spark バージョン 3.5.6 以降の Spark エンジンでのみ使用できます。
- マテリアライズドビューは、最終的にベーステーブルと一致します。更新ウィンドウ中に、マテリアライズドビューに直接アクセスするクエリが古いデータを返す場合があります。現在のデータにすぐにアクセスするには、手動更新を実行します。
- 自動更新の最小間隔は 1 時間です。より頻繁な更新を必要とするユースケースでは、 `REFRESH MATERIALIZED VIEW` コマンドを使用して手動更新をプログラムで実行します。
- クエリの書き換えでは、パフォーマンスよりも正確性が優先されます。マテリアライズドビューが古くなった場合、またはクエリ要件を正確に満たすことができない場合、Spark エンジンはベーステーブルに対して元のクエリを実行します。

Lake Formation でのワークフローを使用したデータのインポート

を使用すると AWS Lake Formation、ワークフローを使用してデータをインポートできます。ワークフローは、データレイクにデータをインポートするためのデータソースとスケジュールを定義します。これは、データレイクのロードとアップデートのプロセスをオーケストレートするために使用される、AWS Glue クローラ、ジョブ、およびトリガーのコンテナです。

トピック

- Lake Formation のブループリントとワークフロー

- ワークフローの作成
- ワークフローの実行

Lake Formation のブループリントとワークフロー

ワークフローは、複雑なマルチジョブの抽出、変換、ロード (ETL) アクティビティをカプセル化します。ワークフローは AWS Glue 、クローラ、ジョブ、トリガーを生成して、データのロードと更新を調整します。Lake Formation は、ワークフローを単一のエンティティとして実行し、追跡します。ワークフローは、オンデマンドで、またはスケジュールに従って実行されるように設定できます。

> **Note**
>
> Spark Parquet ライターは、列名の特殊文字をサポートしていません。これはライター自体の技術的な制限であり、設定の問題ではありません。

Lake Formation で作成するワークフローは、AWS Glue コンソールに DAG (Directed Acyclic Graph) として表示されます。各 DAG ノードは、ジョブ、クローラ、またはトリガーです。進捗状況のモニタリングとトラブルシューティングを行うために、ワークフロー内の各ノードのステータスを追跡することができます。

Lake Formation ワークフローが完了すると、ワークフローを実行したユーザーには、ワークフローが作成する Data Catalog テーブルに対する Lake Formation の SELECT 許可が付与されます。

ワークフローは AWS Glue で作成することもできますが、Lake Formation ではブループリントからワークフローを作成できるため、Lake Formation でのワークフローの作成は、よりシンプルで、自動的です。Lake Formation は、以下のタイプのブループリントを提供します。

- [Database snapshot] (データベーススナップショット) – すべてのテーブルからのデータを、JDBC ソースからデータレイクにロードまたは再ロードします。除外パターンに基づいて、一部のデータをソースから除外することができます。
- [Incremental database] (増分データベース) – 以前に設定されたブックマークに基づいて、新しいデータだけを JDBC ソースからデータレイクにロードします。これに含める JDBC ソースデータベース内の個々のテーブルは、ユーザーが指定します。ブックマーク列とブックマークのソート順をテーブルごとに選択して、以前にロードされたデータを把握しておきます。一連のテーブルに対して増分データベースブループリントを初めて実行すると、ワークフローがそれらのテーブルからすべてのデータをロードして、次回の増分データベースブループリントの実行のためにブックマー

クを設定します。このため、データソース内の各テーブルをパラメータとして指定しておけば、データベーススナップショットブループリントではなく、増分データベースブループリントを使用して、すべてのデータをロードすることができます。

- ログファイル – Elastic Load Balancing ログや Application Load Balancer AWS CloudTrailログなどのログファイルソースからデータを一括ロードします。

以下の表を使用して、データベーススナップショットと増分データベースブループリントのどちらを使用するかを決定してください。

データベーススナップショットを使用する状況	増分データベースを使用する状況
• スキーマ進化に柔軟性がある。(列の名前が変更され、以前の列が削除されて、削除された列の代わりに新しい列が追加される。) • ソースとロード先の間で完全な整合性が必要。	• スキーマ進化が増分的。(列の連続的な追加のみ。) • 新しい行のみが追加され、以前の行は更新されない。

> **Note**
>
> Lake Formation によって作成されたブループリントとワークフローを編集することはできません。

ワークフローの作成

開始する前に、LakeFormationWorkflowRole ロールに必要なデータ許可とデータロケーション許可が付与されていることを確認してください。これは、ワークフローが Data Catalog にメタデータテーブルを作成し、Amazon S3 内のターゲットロケーションにデータを書き込むことができるようにするためです。詳細については、「(オプション) ワークフロー用の IAM ロールを作成する」および「Lake Formation 許可の概要」を参照してください。

> **Note**
>
> Lake Formation は、GetTemplateInstance、GetTemplateInstances、および InstantiateTemplate オペレーションを使用して、ブループリントからワークフローを作成します。これらのオペレーションは一般には公開されておらず、ユーザーに代わってリ

ソースを作成するために内部でのみ使用されます。ユーザーは、ワークフローを作成するための CloudTrail イベントを受け取ります。

ブループリントからワークフローを作成する

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者として、またはデータエンジニア許可を持つユーザーとしてサインインします。詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。
2. ナビゲーションペインで [Blueprints] (ブループリント) を選択してから、[Use blueprint] (ブループリントを使用) を選択します。
3. [Use a blueprint] (ブループリントの使用) ページで、ブループリントタイプを選択するタイルを選択します。
4. [Import source] (インポートソース) で、データソースを指定します。

   JDBC ソースからインポートしている場合は、以下を指定します。

   - [Database connection] (データベース接続) – リストから接続を選択します。AWS Glue コンソールを使用して、追加の接続を作成します。接続の JDBC ユーザー名とパスワードによって、ワークフローがアクセスできるデータベースオブジェクトが決まります。
   - [Source data path] (ソースデータパス) – データベース製品に応じて、*<database>*/*<schema>*/*<table>*、または *<database>*/*<table>* を入力します。Oracle データベース と MySQL は、パス内のスキーマをサポートしません。*<schema>* または *<table>* は、パーセント (%) 文字に置き換えることができます。例えば、システム識別子 (SID) が `orcl` の Oracle データベースの場合は、`orcl/%` を入力して、接続で指定されているユーザーがアクセスできるすべてのテーブルをインポートします。

     > **Important**
     >
     > このフィールドでは、大文字と小文字が区別されます。いずれかのコンポーネントで大文字と小文字の不一致がある場合は、ワークフローが失敗します。

   MySQL データベースを指定した場合、 AWS Glue ETL はデフォルトで Mysql5 JDBC ドライバーを使用するため、MySQL8 はネイティブではサポートされていません。「AWS Glue デベロッパーガイド」の「JDBC connectionType の値」で説明されているよう

に、customJdbcDriverS3Path パラメータを使用するように ETL ジョブスクリプトを編集して、MySQL8 をサポートする別の JDBC ドライバーを使用することができます。

ログファイルからインポートしている場合は、ワークフローに指定するロール (「ワークフローロール」) に、データソースへのアクセスに必要な IAM 許可があることを確認してください。たとえば、 AWS CloudTrail ログをインポートするには、ワークフローの作成中に CloudTrail ログのリストを表示するための cloudtrail:DescribeTrailsおよび アクセスcloudtrail:LookupEvents許可がユーザーに必要です。また、ワークフローロールには Amazon S3 の CloudTrail ロケーションに対するアクセス許可が必要です。

5. 次のいずれかを行います。

- [Database snapshot] (データベーススナップショット) のブループリントタイプの場合は、オプションで、1 つ、または複数の除外パターンを指定することによってインポートするデータのサブセットを特定します。これらの除外パターンは、Unix スタイルの glob パターンです。これらは、ワークフローによって作成されるテーブルのプロパティとして保存されます。

  利用可能な除外パターンの詳細については、「AWS Glue デベロッパーガイド」の「包含パターンと除外パターン」を参照してください。

- [Incremental database] (増分データベース) のブループリントタイプの場合は、以下のフィールドを指定します。インポートするテーブルごとに行を追加してください。

  [Table name] (テーブル名)

  インポートするテーブル。すべて小文字にする必要があります。

  [Bookmark keys] (ブックマークキー)

  ブックマークキーを定義する列名のカンマ区切りのリスト。空白になっている場合は、新しいデータの判別にプライマリキーが使用されます。各列の大文字と小文字は、データソースで定義されている大文字と小文字と一致する必要があります。

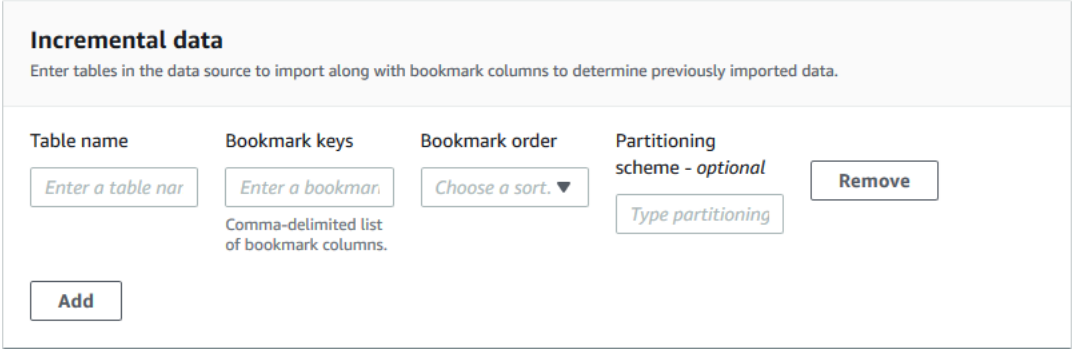
  > **Note**
  >
  > プライマリキーがデフォルトのブックマークキーとして認められるのは、それがギャップを生じることなく連続的に増加または減少している場合のみです。プライマリキーをブックマークキーとして使用したいが、ギャップがあるという場合は、プライマリキー列をブックマークキーとして指定する必要があります。

[Bookmark order] (ブックマークの順序)

[Ascending] (昇順) を選択すると、ブックマークされた値よりも大きい値を持つ行が新しい行として識別されます。[Descending] (降順) を選択すると、ブックマークされた値より小さい値を持つ行が新しい行として識別されます。

[Partitioning scheme] (パーティショニングスキーム)

(オプション) スラッシュ (/) で区切られた、パーティショニングキー列のリスト。例えば、`year/month/day` などです。



詳細については、「AWS Glue デベロッパーガイド」の「ジョブのブックマークを使用した処理済みデータの追跡」を参照してください。

6. [Import target] (インポートターゲット) で、ターゲットデータベース、ターゲット Amazon S3 ロケーション、およびデータ形式を指定します。

   ワークフローロールに、データベースと Amazon S3 ターゲットロケーションに対する必要な Lake Formation 許可があることを確認してください。

   > **Note**
   >
   > 現在、ブループリントはターゲットでのデータの暗号化をサポートしていません。

7. インポートの頻度 を選択します。

   [Custom] (カスタム) オプションでは、`cron` 式を指定することができます。

8. [Import options] (インポートオプション) で以下を実行します。

   a. ワークフロー名を入力します。

b. ロールには、「(オプション) ワークフロー用の IAM ロールを作成する」で作成したロール `LakeFormationWorkflowRole` を選択します。

c. オプションで、テーブルプレフィックスを指定します。プレフィックスは、ワークフローが作成する Data Catalog テーブルの名前の前に付加されます。

9. [Create] (作成) を選択し、ワークフローが正常に作成されたことコンソールが報告するまで待機します。

> **Tip**
>
> 以下のエラーメッセージが表示されましたか?
> `User: arn:aws:iam::<account-id>:user/<username> is not authorized to perform: iam:PassRole on resource:arn:aws:iam::<account-id>:role/<rolename>...`
> その場合は、*<account-id>* をすべてのポリシーで有効な AWS アカウント番号に置き換えたことを確認します。

> 以下も参照してください。
>
> - Lake Formation のブループリントとワークフロー

ワークフローの実行

ワークフローは、Lake Formation コンソール、AWS Glue コンソール、AWS Glue コマンドラインインターフェイス (AWS CLI)、または API を使用して実行することができます。

ワークフローを実行する (Lake Formation コンソール)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者として、またはデータエンジニア許可を持つユーザーとしてサインインします。詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。
2. ナビゲーションペインで [Blueprints] (ブループリント) を選択します。
3. [Blueprints] (ブループリント) ページで、ワークフローを選択します。次に、[Actions] (アクション) メニューで [Start] (開始) を選択します。

4. ワークフローの実行に伴って、その進捗状況を [Last run status] (最終実行ステータス) 列で確認します。更新ボタンを随時選択します。

   ステータスは、[RUNNING] (実行中) から、[Discovering] (検出中)、[Importing] (インポート中)、[COMPLETED] (完了) と移行します。

   ワークフローが完了すると、以下のようになります。

   - Data Catalog に新しいメタデータテーブルがある。
   - データがデータレイクに取り込まれる。

   ワークフローが失敗する場合は、以下を実行します。

   a. ワークフローを選択します。[Actions] (アクション) を選択してから、[View graph] (グラフを表示) を選択します。

      AWS Glue コンソールでワークフローが開きます。

   b. そのワークフローが選択されていることを確認し、[History] (履歴) タブを選択します。

   c. [History] (履歴) で、最新の実行を選択し、[View run details] (実行の詳細を表示) を選択します。

   d. 動的 (ランタイム) グラフで失敗したジョブまたはクローラを選択し、エラーメッセージを確認します。障害が発生したノードは赤色または黄色のいずれかになっています。

以下も参照してください。

- Lake Formation のブループリントとワークフロー

へのデータの取り込み AWS Glue Data Catalog

AWS Glue Data Catalog (データカタログ) でフェデレーティッドカタログを作成し、Amazon S3 データレイクと Amazon Redshift データウェアハウス間でデータを統合できます。また、 などの運用データベースや、PostgreSQL Amazon DynamoDB、Google BigQuery、MySQL などのサードパーティーデータソースからのデータを統合することもできます。データカタログは、一元化されたメタデータリポジトリを提供し、異種システム間でのデータの管理と発見を容易にします。

データカタログは、フェデレーティッドコネクタを介して 30 個を超える外部データソースと統合されます。この統合により、 AWS データパイプラインを構築して最初にデータを に取り込むことなく、これらの外部ソースからデータをクエリできます。

外部データをカタログ化した後、 AWS Lake Formation を使用して Data Catalog 内のデータアクセス許可を一元管理できます。データレイク管理者は、タグベースのアクセスコントロール (LF タグ) と名前付きリソースメソッドを使用して、同じアカウント内またはアカウント間で、他の IAM プリンシパル (ユーザーまたはロール) にきめ細かなアクセス許可を付与できます。

LF タグを使用すると、データ管理者はドメインや機密性レベルなどの属性に基づいてリソースを論理的に整理できるため、アクセス許可管理を簡素化しながら、Athena、Amazon EMR、 AWS Glue Redshift Spectrum などの分析および機械学習サービス間で一貫したアクセスコントロールを確保できます。

データカタログは、外部データセットと外部メタストアのデータとアクセス許可を管理するために、次の方法を提供します。

- Amazon S3 Table バケットを Data Catalog と統合する – Amazon S3 Tables を Data Catalog オブジェクトとして公開およびカタログ化し、Lake Formation コンソールまたは AWS Glue API オペレーションを使用して、カタログを Lake Formation データの場所として登録できます。
- Amazon Redshift データウェアハウスにデータを取り込む – AWS Glue Data Catalog既存の Amazon Redshift 名前空間またはクラスターをデータカタログに登録し、データカタログにマルチレベルフェデレーティッドカタログを作成します。

  Amazon EMR Serverless や Amazon Athena などの、Apache Iceberg REST カタログ OpenAPI 仕様と互換性のある任意のクエリエンジンを使用してデータにアクセスできます。
- リモートIceberg RESTカタログをデータカタログにフェデレーションする – リモートIceberg RESTカタログをデータカタログにフェデレーションし、 AWS 分析エンジンを使用して Amazon S3 に保存されているリモートIcebergテーブルに安全にアクセスします。

- 外部データソースから Data Catalog にフェデレーションする – AWS Glue 接続を使用して Data Catalog を外部データソースに接続し、フェデレーションカタログを作成して Lake Formation を使用してデータセットへのアクセス許可を一元管理します。データカタログへのメタデータの移行は不要です。
- データカタログで Amazon Redshift テーブルを管理するカタログを作成する – 現在、Amazon Redshift プロデューサークラスターや Amazon Redshift データ共有は利用できないが、データカタログで Amazon Redshift テーブルを作成して管理する必要がある場合があります。`glue:CreateCatalog` API オペレーションまたは AWS Lake Formation コンソールを使用して AWS Glue マネージドカタログを作成し、カタログタイプを `Managed` とし、`Catalog source` を Redshift と設定することで開始できます。
- Amazon Redshift データ共有をデータカタログで公開する – Amazon Redshift データ共有をデータカタログに公開し、Lake Formation を使用してデータ共有のデータアクセスを一元管理し、ユーザーアクセスを制限します。

  Amazon Redshift Spectrum を使用したデータのクエリ
- データカタログを外部 Hive メタストアに接続する - データカタログを外部メタストアに接続し、Lake Formation を使用して Amazon S3 のデータセットに対するアクセス許可を管理します。データカタログへのメタデータの移行は不要です。
- Lake Formation と AWS Data Exchange の統合 – Lake Formation は、 を介したデータへのアクセスのライセンスをサポートしています AWS Data Exchange。Lake Formation データのライセンスを付与する場合は、「AWS Data Exchange ユーザーガイドの「AWS Data Exchangeとは」を参照してください。

トピック

- リモート Iceberg カタログへのカタログフェデレーション
- Amazon Redshift データを に取り込む AWS Glue Data Catalog
- の外部データソースへのフェデレーション AWS Glue Data Catalog
- Amazon S3 Tables と AWS Glue Data Catalog および の統合 AWS Lake Formation
- S3 Tables 統合のアクセスコントロールの変更
- での Amazon Redshift マネージドカタログの作成 AWS Glue Data Catalog
- Amazon Redshift データ共有でのデータに対するアクセス許可の管理
- 外部メタストアを使用するデータセットのアクセス許可の管理

リモート Iceberg カタログへのカタログフェデレーション

のカタログフェデレーション AWS Glue は、 AWS 分析エンジンを使用して、Amazon S3 に保存され、リモートカタログにカタログ化された Iceberg テーブルへの直接かつ安全なアクセスを提供します。カタログフェデレーションは、リモートテーブルにアクセスするときに、データカタログとリモートカタログ間でメタデータを同期します。Amazon Redshift、Amazon EMR、Amazon Athena、Apache Spark などのサードパーティーエンジンなど AWS Glue、さまざまな分析エンジンでサポートされています。

カタログフェデレーションは AWS Glue Data Catalog 、 を使用してリモートカタログシステムと通信してテーブルを検出し、Lake Formation を使用して Amazon S3 のテーブルデータへのアクセスを許可します。フェデレーティッドテーブルをクエリすると、Data Catalog はクエリ時にリモートカタログ内の最新のテーブル情報を検出し、テーブルの Amazon S3 の場所、現在のスキーマ、パーティション情報を取得します。その後、分析エンジン (Amazon Athena、Amazon Redshift、Amazon EMR) はこの情報を使用してAmazon S3から直接 Iceberg データファイルにアクセスします。Lake Formation は、Amazon S3 に保存されているテーブルデータにスコープ付き認証情報を提供することでテーブル (複数可) へのアクセスを管理し、エンジンがフェデレーティッドテーブル (複数可) にきめ細かなアクセス許可を適用できるようにします。

カタログフェデレーションの機能

Lake Formation を使用して管理

Data Catalog のフェデレーティッド Iceberg カタログは Lake Formation 登録リソースであり、Lake Formation 許可を使用してフェデレーティッド Iceberg カタログの Iceberg テーブルにきめ細かな行、列、セルレベルのアクセス許可を付与できます。フェデレーティッド Iceberg カタログと関連オブジェクトは、 AWS アカウント間で安全に共有できます。フェデレーティッド Iceberg カタログは Lake Formation タグベースのアクセスコントロールとも連携するため、タグを使用してガバナンスをスケーリングできます。

ネットワーク設定

カタログフェデレーションは、標準の HTTPS 接続を使用したリモートカタログソースへの直接接続をサポートします。また、ネットワークの分離を維持する場合は Amazon VPC 経由の接続をサポートし、組織のファイアウォール経由の安全な通信が必要な場合はプロキシサポート経由の接続もサポートします。

制限事項

- Iceberg テーブルメタデータサイズ制限 – フェデレーティッド Iceberg カタログの場合、 は REST API コールごとに最大 5 MB のメタデータサイズ AWS Glue Data Catalog をサポートします。 はこの制限を超えるメタデータを持つテーブルへのリクエスト AWS Glue Data Catalog を拒否します。

トピック

- Snowflake Iceberg Catalog へのフェデレーション
- へのフェデレーション Databricks Unity Catalog

Snowflake Iceberg Catalog へのフェデレーション

AWS Glue Data Catalog は、Snowflakeサービスプリンシパルの OAuth2 認証情報Snowflakeを使用して にフェデレーションします。このフェデレーションを使用して、 Snowflake HorizonおよびSnowflake Polarisカタログに接続できます。この認証メカニズムにより、Data Catalog はサービスプリンシパルに関連付けられた権限に基づいて、カタログ内のさまざまなオブジェクト (カタログ、データベース、テーブルなど) のメタデータにアクセスできます。適切なオブジェクトへのアクセスを確保するには、これらのオブジェクトのメタデータを読み取るために必要なアクセス許可Snowflakeを のサービスプリンシパルに付与することが重要です。

前提条件

Lake Formation によって管理されるフェデレーティッドカタログを Data Catalog で作成する前に、次のアクセス許可があることを確認してください。

IAM プリンシパル (ユーザーまたはロール) には、次のアクセス許可が必要です。

- Lake Formation のアクセス許可 – `lakeformation:RegisterResource`、`lakeformation:DescribeResource`
- アクセスAWS Glue 許可 – `glue:CreateConnection`、`glue:CreateCatalog`、`glue:GetConnection`、`glue:PassConnection`
- Secrets Manager のアクセス許可 – `secretsmanager:CreateSecret`、`secretsmanager:GetSecretValue`
- IAM アクセス許可 – `iam:CreateRole`、`iam:AttachRolePolicy`、 `iam:PassRole`

Lake Formation データレイク管理者であるか、データカタログに対するCREATE_CATALOGアクセス許可を持っている必要があります

フェデレーティッドカタログの作成

コンソールの使用

1. コンソールにサインインし、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールを開きます。
2. AWS リージョンセレクタで任意のリージョンを選択します。
3. 左側のナビゲーションペインで、[カタログ] を選択します。
4. Create Catalog を選択して、Create Catalog ワークフローを開きます。
5. データソースの選択ステップで、使用可能なオプションSnowflakeから を選択します。
6. カタログの詳細の設定ステップでは、カタログの詳細、接続の詳細、登録の詳細の 3 つの情報を指定します。
7. カタログ詳細コンテナで、フェ AWS Glue デレーティッドカタログに一意の名前を指定し、既存のSnowflakeカタログの名前を入力します。
8. 接続の詳細コンテナでは、アクセスできる既存の接続から選択するか、新しいコネクタを作成するための設定を指定できます。
9. 新しい接続設定は次のとおりです。

   - 接続名 – AWS Glue 接続オブジェクトの一意の名前。
   - インスタンス URL – 既存のSnowflakeアカウントのエンドポイント URL。
   - 認証 – がリモートカタログサーバーへの接続 AWS Glue に使用する認証設定を指定します。は OAuth2 認証とカスタム認証の両方 AWS Glue をサポートします。
   - トークン URL – リモートカタログの ID プロバイダーの URL を指定します。
   - OAuth2 クライアント ID – (オプション) リモートカタログに関連付けられた OAuth2 認証情報のクライアント ID を指定します。クライアント ID を指定しない場合、トークンリクエストは に保存されているクライアントシークレットのみを使用して認証します AWS Secrets Manager。
   - シークレット – を使用して OAuth2 クライアントシークレットを保存して使用する AWS Secrets Manager か、テキストボックスにシークレット値を入力します。コンソールでシークレットを手動で入力すると、 はユーザーに代わってシークレット AWS Glue を作成します。
   - トークン URL スコープ – 認証の OAuth スコープを指定します。

- カタログケーシングフィルター – リモートカタログから Data Catalog に小文字または大文字のオブジェクトを持ち込むかどうかを選択します。

10. AWS Glue および Lake Formation サービスプリンシパルがそれぞれリモート Iceberg テーブルの AWS Secrets Manager および Amazon S3 ロケーションのシークレットにアクセスするために使用できる IAM ロールを作成します。登録ドロップダウンで IAM ロールを選択します。IAM ポリシーの詳細については、次の CLI セクションのステップ 2 と 3 を参照してください。
11. 接続のテストを選択して、接続プロパティと IAM ロールアクセスが正しく設定されているかどうかをテストします。
12. 次へ を選択して設定を確認します。
13. レビューページでカタログの作成を選択します。

CLI を使用する

1. AWS Secrets Manager シークレットを作成する

   AWS Glue コネクタは、OAuth2 と Custom の 2 つの認証タイプをサポートしています。OAuth2 オプションを使用する場合は、 AWS Secrets Manager を使用してSnowflakeサービスプリンシパルのクライアントシークレットを保存します。このシークレットは、後で AWS Glue 接続を作成するときに使用します。カスタム認証では、 AWS Secrets Manager を使用してアクセストークンを保存および取得します。

   次の例では、*your-snowflake-secret*、 *client_secretregion*を独自の情報に置き換えます。

```
aws secretsmanager create-secret \
--name your-snowflake-secret \
--description "Snowflake secret" \
--secret-string '{
"USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET": "client_secret"
}' \
--region region
```

   > **Note**
   >
   > USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET は、 AWS Glue がシークレット内のクライアントシークレット値を参照するために使用する予約キーワードで

す。Lake Formation コンソールでシークレットを作成するときにも、同じキーワードを使用します。

2. 前のステップで作成したシークレットへのアクセスを AWS Glue 接続オブジェクトに許可する IAM ロールを作成する

   AWS Glue 接続オブジェクトでは、 AWS Secrets Manager を使用して OAuth AWS Secrets Manager シークレットトークンを保存、取得、更新するときに、シークレットにアクセスする必要があります。 AWS Glue 接続オブジェクトには、Amazon VPC エンドポイントを使用して Snowflakeアカウントへの接続を制限するときに、Amazon VPC ネットワークインターフェイスを作成、説明、使用するためのアクセスも必要です。

   IAM ポリシーを作成し、IAM ロールにアタッチします。信頼ポリシーに AWS Glue サービスプリンシパルを追加します。

   次の例では、*your-secrets-manager-ARN*、、 *your-vpc-idyour-subnet-id1*を独自の情報に置き換えます。

   Example IAM ポリシー

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:PutSecretValue"
            ],
            "Resource": [
                "your-secrets-manager-ARN"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "ArnEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account-id:vpc/your-vpc-id",
                    "ec2:Subnet": ["arn:aws:ec2:region:account-id:subnet/your-
subnet-id1"]
                }
            }
        }
    ]
}
```

Example信頼ポリシー

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

3. Lake Formation にカタログの Amazon S3 ロケーションへの読み取りアクセスを許可する IAM ポリシーを作成する

   Data Catalog のフェデレーティッドカタログのカタログ所有者は、Lake Formation を使用して、データチームへの粗粒度のテーブルアクセス、粒度の細かい列レベル、行レベル、セルレベルのアクセス、タグベースのアクセスを付与します。Lake Formation は、リモート Iceberg テーブルの基盤となる Amazon S3 ロケーションへのアクセスを許可する IAM ロールを使用します。このアクセスにより、Lake Formation はリモートテーブルをクエリする分析エンジンにスコープ付きアクセス認証情報を提供できます。

   IAM ポリシーを作成し、IAM ロールにアタッチします。Lake Formation サービスプリンシパルをロールの信頼ポリシーに追加します。

   以下の例では、*amzn-s3-demo-bucketN* と *your-kms-key* を独自の情報に置き換えます。

   Example IAM ポリシー

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                "arn:aws:s3:::amzn-s3-demo-bucket2/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt"
            ],
            "Resource": [
                "your-kms-key"
            ]
        }
    ]
}
```

Example信頼ポリシー

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
            "Service": "lakeformation.amazonaws.com"
```

```
        },
        "Action": "sts:AssumeRole"
    }]
}
```

> **Note**
>
> Lake Formation コンソールを使用してフェデレーティッドカタログを作成する場合、コンソールは両方のポリシーがアタッチされた単一の IAM ロールを使用してセットアップを完了します。

4. AWS Glue 接続オブジェクトを作成する

AWS Glue コネクタは、OAuth2 およびカスタム認証方法をサポートしています。Data Catalog 接続オブジェクトは、リモートカタログ内の小文字または大文字のオブジェクトを Data Catalog に持ち込むことができるCATALOG_CASING_FILTER設定をサポートしています。

次の例では、OAuth2 認証設定を使用して AWS Glue 接続を作成します。*highlighted sections* を自分の情報に置き換えます。

```
aws glue create-connection \
    --connection-input '{
"Name": "your-glue-connection-to-snowflake-account",
"ConnectionType": "SNOWFLAKEICEBERGRESTCATALOG",
"ConnectionProperties": {
    "INSTANCE_URL": "your-snowflake-account-URL",
    "ROLE_ARN": "your-IAM-role-for-secrets-and-VPC-access",
    "CATALOG_CASING_FILTER": "LOWERCASE_ONLY"
},
"AuthenticationConfiguration": {
    "AuthenticationType": "OAUTH2",
    "OAuth2Properties": {
        "OAuth2GrantType": "CLIENT_CREDENTIALS",
        "TokenUrl": "your-internal-or-external-token-server-url",
        "OAuth2ClientApplication": {
            "UserManagedClientApplicationClientId": "our-client-id"
        },
        "TokenUrlParametersMap": {
            "scope": "all-apis"
        }
    },
```

```
    "SecretArn": "arn:aws:secretsmanager:your-aws-region:your-aws-account-
id:secret:snowflake-secret"
}
}'
```

> **Note**
>
> 0Auth2ClientApplication フィールドはオプションです。これを省略すると、トークンリクエストは、クライアント ID を送信せずに AWS Secrets Manager、 に保存されているクライアントシークレットのみを使用して認証されます。

5. Lake Formation リソースとして AWS Glue 接続を登録する

   AWS Glue 接続オブジェクト (ステップ 4 で作成) と IAM ロール (ステップ 3 で作成) を使用して AWS Glue 、接続オブジェクトを Lake Formation マネージドリソースとして登録できるようになりました。

   your-glue-connector-arn と *your-IAM-role-ARN-having-LF-access* をあなたの情報に置き換えます。

   ```
   aws lakeformation register-resource \
       --resource-arn your-glue-connector-arn \
       --role-arn your-IAM-role-ARN-having-LF-access \
       --with-federation \
       --with-privileged-access
   ```

6. Data Catalog でフェデレーティッドカタログを作成する

   AWS Glue 接続オブジェクトを作成して Lake Formation に登録したら、データカタログにフェデレーションカタログを作成できます。

   フェデレーティッドカタログに で一意の名前を付け*your-federated-catalog-name*、Snowflakeで カタログを参照し*catalog-name-in-Snowflake*、 で前に作成した入力接続名を指定します*your-glue-connection-name*。

   ```
   aws glue create-catalog \
       --name your-federated-catalog-name \
       --catalog-input '{
       "FederatedCatalog": {
           "Identifier": catalog-name-in-Snowflake",
           "ConnectionName": your-glue-connection-name"
   ```

```
    },
    "CreateTableDefaultPermissions": [],
    "CreateDatabaseDefaultPermissions": []
}'
```

と統合する際の考慮事項 Snowflake

- でリソース (データベースやテーブルなど) を削除してもSnowflake、Lake Formation はそのフェデレーティッドリソースに付与されたアクセス許可を自動的に取り消しません。アクセス許可を削除するには、Lake Formation を使用してフェデレーティッドリソースで以前に付与されたアクセス許可を明示的に取り消す必要があります。
- `CATALOG_CASING_FILTER='UPPERCASE_ONLY'` 設定を使用してリモートカタログをマウントすると、大文字の識別子を持つデータベースとテーブルはフェデレーションされますが、小文字の識別子を持つオブジェクトはフェデレーションされません。
- 同じ AWS Glue 接続を再利用して、複数のフェデレーティッドカタログを作成できます。カタログを削除しても、関連付けられた接続は削除されません。接続を削除するには、 AWS CLI `aws glue delete-connection` コマンドを使用し、関連するすべてのカタログが最初に削除されていることを確認してください。
- Polaris カタログ内のネストされた名前空間はサポートされていません。つまり、カタログフェデレーションは、3 つの部分からなる表記 に従うリモートIcebergテーブルにアクセスできます`catalog.database.table`。

へのフェデレーション Databricks Unity Catalog

AWS Glue Data Catalog は、Databricksサービスプリンシパルの OAuth2 認証情報Databricksを使用して にフェデレーションします。この認証メカニズムにより AWS Glue Data Catalog 、 はサービスプリンシパルに関連付けられた権限に基づいて、 のさまざまなオブジェクト (カタログ、データベースDatabricks Unity Catalog、テーブルなど) のメタデータにアクセスできます。適切なオブジェクトへのアクセスを確保するには、これらのオブジェクトのメタデータを読み取るDatabricksために必要なアクセス許可を のサービスプリンシパルに付与することが重要です。

次に、カタログフェデレーションにより、Databricks Unity Catalog 内のIcebergテーブルの検出とクエリが可能になります。デルタテーブルを読み取るには、 を使用してこれらのテーブルでIcebergメタデータが使用可能であることを確認してくださいUniform。Databricks チュートリアルとドキュメントに従って、Databricksワークスペースにサービスプリンシパルと関連する権限を作成します。

前提条件

Lake Formation によって管理されるフェデレーティッドカタログを Data Catalog で作成する前に、次のアクセス許可があることを確認してください。

IAM プリンシパル (ユーザーまたはロール) には、次のアクセス許可が必要です。

- Lake Formation のアクセス許可 – `lakeformation:RegisterResource`、`lakeformation:DescribeResource`
- アクセスAWS Glue 許可 – `glue:CreateConnection`、`glue:CreateCatalog`、`glue:GetConnection`、`glue:PassConnection`
- Secrets Manager のアクセス許可 – `secretsmanager:CreateSecret`、`secretsmanager:GetSecretValue`
- IAM アクセス許可 – `iam:CreateRole`、`iam:AttachRolePolicy`、 `iam:PassRole`

Lake Formation データレイク管理者であるか、データカタログに対するCREATE_CATALOGアクセス許可を持っている必要があります

フェデレーティッドカタログの作成

コンソールの使用

1. コンソールにサインインし、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールを開きます。
2. AWS リージョンセレクタで任意のリージョンを選択します。
3. 左側のナビゲーションペインで、[カタログ] を選択します。
4. Create Catalog を選択して、Create Catalog ワークフローを開きます。
5. データソースの選択ステップで、使用可能なオプションDatabricksから を選択します。
6. カタログの詳細の設定ステップでは、カタログの詳細、接続の詳細、登録の詳細の 3 つの情報を指定します。
7. カタログ詳細コンテナで、フェ AWS Glue デレーティッドカタログに一意の名前を指定し、既存のDatabricksカタログの名前を入力します。
8. 接続の詳細コンテナでは、アクセスできる既存の接続から選択するか、新しいコネクタを作成するための設定を指定できます。
9. 新しい接続設定は次のとおりです。

- 接続名 – AWS Glue 接続オブジェクトの一意の名前。
- Workspace URL – 既存のDatabricksワークスペースのエンドポイント URL。
- 認証 – がリモートカタログサーバーへの接続 AWS Glue に使用する認証設定を指定します。は OAuth2 認証とカスタム認証の両方 AWS Glue をサポートします。
- トークン URL – リモートカタログの ID プロバイダーの URL を指定します。
- OAuth2 クライアント ID – リモートカタログに関連付けられた OAuth2 認証情報のクライアント ID を指定します。
- シークレット – を使用して OAuth2 クライアントシークレットを保存して使用する AWS Secrets Manager か、テキストボックスにシークレット値を入力します。コンソールでシークレットを手動で入力すると、 はユーザーに代わってシークレット AWS Glue を作成します。
- トークン URL スコープ – 認証の OAuth スコープを指定します。

10. AWS Glue および Lake Formation サービスプリンシパルがそれぞれリモートIcebergテーブルのおよび Amazon S3 ロケーションのシークレットにアクセスするために使用できる IAM ロールを作成します。登録ドロップダウンで IAM ロールを選択します。IAM ポリシーの詳細については、次の CLI セクションのステップ 2 と 3 を参照してください。
11. 接続のテストを選択して、接続プロパティと IAM ロールアクセスが正しく設定されているかどうかをテストします。Amazon VPC Databricksを使用して に接続する場合、テスト接続機能は利用できません。
12. 次へ を選択して設定を確認します。
13. レビューページでカタログの作成を選択します。

CLI を使用する

1. AWS Secrets Manager シークレットを作成する

   AWS Glue コネクタは、OAuth2 とカスタムの 2 つの認証タイプをサポートしています。OAuth2 オプションを使用する場合は、 AWS Secrets Manager を使用してDatabricksサービスプリンシパルのクライアントシークレットを保存します。このシークレットは、後で AWS Glue 接続を作成するときに使用します。カスタム認証の場合は、 AWS Secrets Manager を使用してアクセストークンを保存および取得します。

   次の例では、*<databricks-secret>*、 *<client_secret><region>*を独自の情報に置き換えます。

   ```
   aws secretsmanager create-secret \
   ```

```
--name <databricks-secret> \
--description "Databricks secret" \
--secret-string '{
"USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET": "<client_secret>"
}' \
--region <region>
```

> **Note**
>
> USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET は、 AWS Glue がシークレット内のクライアントシークレット値を参照するために使用する予約キーワードです。Lake Formation コンソールでシークレットを作成するときにも、同じキーワードを使用します。

2. 前のステップで作成したシークレットへのアクセスを AWS Glue 接続オブジェクトに許可する IAM ロールを作成する

   AWS Glue 接続オブジェクトでは、 AWS Secrets Manager を使用して OAuth AWS Secrets Manager シークレットトークンを保存、取得、更新するときに、シークレットにアクセスする必要があります。 AWS Glue 接続オブジェクトには、Amazon VPC エンドポイントを使用して Databricksワークスペースへの接続を制限するときに、Amazon VPC ネットワークインターフェイスを作成、説明、使用するためのアクセスも必要です。

   IAM ポリシーを作成し、IAM ロールにアタッチします。信頼ポリシーに AWS Glue サービスプリンシパルを追加します。

   次の例では、*<your-secrets-manager-ARN>*、、 *<your-vpc-id><your-subnet-id1>*を独自の情報に置き換えます。

   Example IAM ポリシー

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:PutSecretValue"
            ],
```

```
            "Resource": [
                "<your-secrets-manager-ARN>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account-id:vpc/<your-vpc-id>",
                    "ec2:Subnet": ["arn:aws:ec2:region:account-id:subnet/<your-
subnet-id1>"]
                }
            }
        }
    ]
}
```

Example信頼ポリシー

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

3. Lake Formation にカタログの Amazon S3 ロケーションへの読み取りアクセスを許可する IAM ポリシーを作成する

   Data Catalog のフェデレーティッドカタログのカタログ所有者は、Lake Formation を使用して、データチームへの粗粒度のテーブルアクセス、粒度の細かい列レベル、行レベル、セルレベルのアクセス、タグベースのアクセスを付与します。Lake Formation は、リモート Iceberg

テーブルの基盤となる Amazon S3 ロケーションへのアクセスを許可する IAM ロールを使用します。このアクセスにより、Lake Formation はリモートテーブルをクエリする分析エンジンにスコープ付きアクセス認証情報を提供できます。

IAM ポリシーを作成し、IAM ロールにアタッチします。Lake Formation サービスプリンシパルを IAM ロールの信頼ポリシーに追加します。

以下の例では、*<your-s3-bucket-N>* と *<your-kms-key>* を独自の情報に置き換えます。

Example IAM ポリシー

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<your-s3-bucket-1>/*",
                "arn:aws:s3:::<your-s3-bucket-2>/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<your-s3-bucket-1>",
                "arn:aws:s3:::<your-s3-bucket-2>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt"
            ],
            "Resource": [
                "<your-kms-key>"
            ]
        }
```

```
    ]
}
```

Example信頼ポリシー

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

> **Note**
>
> Lake Formation コンソールを使用してフェデレーティッドカタログを作成する場合、コンソールは両方のポリシーがアタッチされた単一の IAM ロールを使用してセットアップを完了します。

4. AWS Glue 接続オブジェクトを作成する

Data Catalog は、Data Catalog を に接続DATABRICKSICEBERGRESTCATALOGするための connectionType: をサポートしていますDatabricks。この AWS Glue コネクタは、OAuth2 および カスタム認証方法をサポートしています。

次の例では、OAuth2 認証設定を使用して AWS Glue 接続を作成します。*highlighed sections* を自分の情報に置き換えます。

```
aws glue create-connection \
    --connection-input '{
"Name": "<your-glue-connection-to-databricks-unity-account>",
"ConnectionType": "DATABRICKSICEBERGRESTCATALOG",
"ConnectionProperties": {
    "INSTANCE_URL": "<your-databricks-workspace-catalog-URL>",
    "ROLE_ARN": "<your-IAM-role-for-secrets-and-VPC-access>"
},
"AuthenticationConfiguration": {
```

```
    "AuthenticationType": "OAUTH2",
    "OAuth2Properties": {
        "OAuth2GrantType": "CLIENT_CREDENTIALS",
        "TokenUrl": "<your-internal-or-external-token-server-url>",
        "OAuth2ClientApplication": {
            "UserManagedClientApplicationClientId": "<your-client-id>"
        },
        "TokenUrlParametersMap": {
            "scope": "all-apis"
        }
    },
    "SecretArn": "arn:aws:secretsmanager:<aws-region>:<your-aws-account-
id>:secret:<databricks-secret>"
}
}'
```

5. Lake Formation リソースとして AWS Glue 接続を登録する

   AWS Glue 接続オブジェクト (ステップ 4 で作成) と IAM ロール (ステップ 3 で作成) を使用して、 AWS Glue 接続オブジェクトを Lake Formation マネージドリソースとして登録できるようになりました。

   <your-glue-connector-arn> と *<your-IAM-role-ARN-having-LF-access>* をあなたの情報に置き換えます。

   ```
   aws lakeformation register-resource \
       --resource-arn <your-glue-connector-arn> \
       --role-arn <your-IAM-role-ARN-having-LF-access> \
       --with-federation \
       --with-privileged-access
   ```

6. Data Catalog でフェデレーティッドカタログを作成する

   AWS Glue 接続オブジェクトを作成して Lake Formation に登録したら、データカタログにフェデレーションカタログを作成できます。

   フェデレーティッドカタログに で一意の名前を付け*<your-federated-catalog-name>*、Databricksの カタログを参照し*"<catalog-name-in-Databricks>*、 で前に作成した入力接続名を指定します*"<your-glue-connection-name>*。

   ```
   aws glue create-catalog \
       --name <your-federated-catalog-name> \
   ```

```
    --catalog-input '{
    "FederatedCatalog": {
        "Identifier": "<catalog-name-in-Databricks>",
        "ConnectionName": "<your-glue-connection-name>"
    },
    "CreateTableDefaultPermissions": [],
    "CreateDatabaseDefaultPermissions": []
}'
```

と統合する際の考慮事項 Databricks

- でリソース (データベースやテーブルなど) を削除してもDatabricks、Lake Formation はそのフェデレーティッドリソースに付与されたアクセス許可を自動的に取り消しません。アクセス許可を削除するには、Lake Formation を使用してフェデレーティッドリソースで以前に付与されたアクセス許可を明示的に取り消す必要があります。
- この統合を使用して、Amazon S3 に保存されている Iceberg テーブルをクエリできます。他のテーブル形式またはオブジェクトストレージを使用する場合、リモートカタログのメタデータを にフェデレーション AWS Glue してデータベースとテーブルを一覧表示できますが、 のようなクエリオペレーション`SELECT ColumnFoo from TableBar`はクエリ中に失敗し、Apache Iceberg「テーブルの読み取りに失敗しました」というエラーが発生します。オブジェクトストレージの場所はサポートされていません。
- 同じ AWS Glue 接続を再利用して、複数のフェデレーティッドカタログを作成できます。カタログを削除しても、関連付けられた接続オブジェクトは削除されません。接続オブジェクトを削除するには、 AWS CLI `aws glue delete-connection` コマンドを使用し、関連するすべてのカタログが最初に削除されていることを確認してください。

Amazon Redshift データを に取り込む AWS Glue Data Catalog

AWS Glue Data Catalog (データカタログ) の Amazon Redshift データウェアハウスで分析データを管理し、Amazon S3 データレイクと Amazon Redshift データウェアハウスを統合できます。Amazon Redshift は、 AWS クラウド内のフルマネージド型のペタバイト規模のデータウェアハウスサービスです。Amazon Redshift データウェアハウスは、ノードと呼ばれるコンピューティングリソースの集合で、クラスターと呼ばれるグループに編成されています。各クラスターは Amazon Redshift エンジンを実行し、1 つ以上のデータベースを含みます。

Amazon Redshift では、Amazon Redshift でプロビジョニングされたクラスターとサーバーレス名前空間を作成し、データカタログに登録できます。これにより、Amazon Redshift マネージドストレー

ジ (RMS) と Amazon S3 バケットのデータを統合し、Apache Iceberg 互換の分析エンジンからデータにアクセスできるようになります。

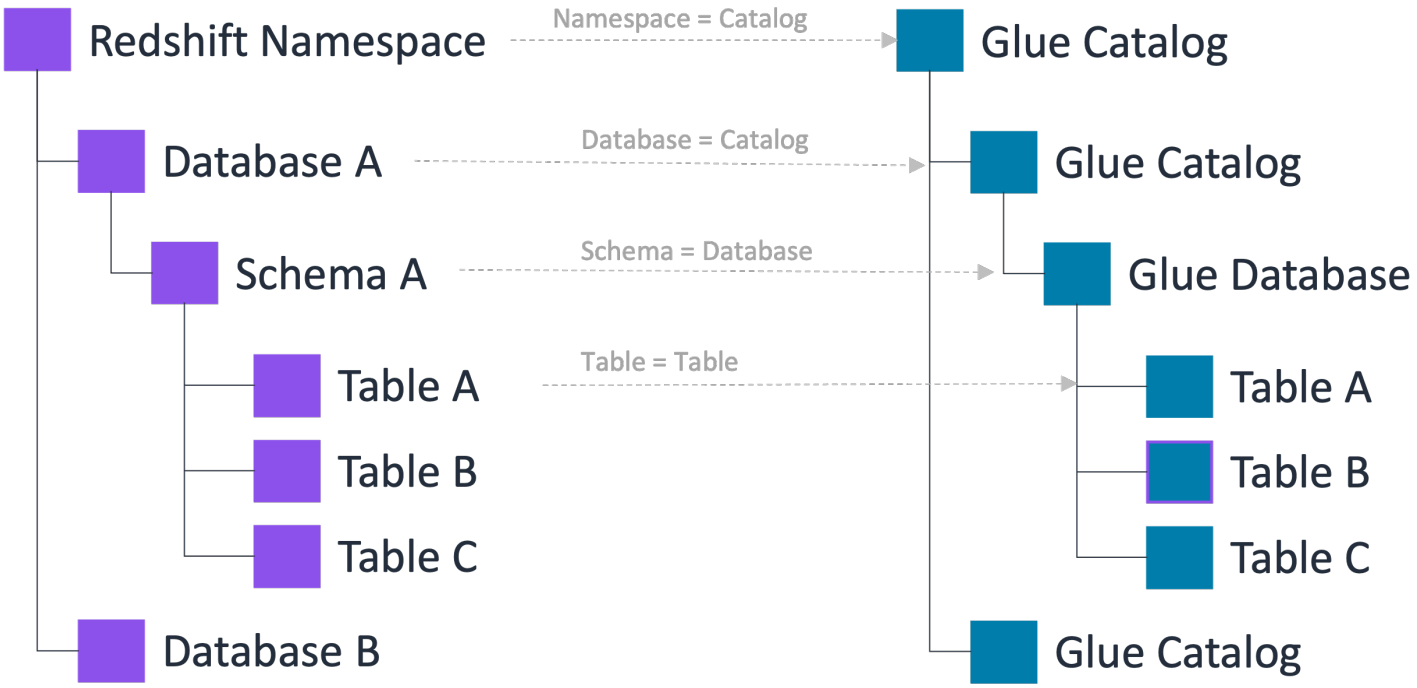
名前空間とクラスターを登録することで、データをコピーまたは移動することなく、データへのアクセスを提供できます。Amazon Redshift でのクラスターと名前空間の登録の詳細については、「Amazon Redshift のクラスターや名前空間の AWS Glue Data Catalogへの登録」を参照してください。

Amazon Redshift では、データ共有を通じて、または名前空間とクラスターをデータカタログに登録することで、データ共有を実行できます。個々のデータベースオブジェクトレベルで動作するデータ共有では、テーブルまたはビューごとに共有を有効にする必要があります。これとは対照的に、名前空間はクラスターまたは名前空間レベルで関数を発行します。クラスターまたは名前空間をデータカタログに登録すると、その中のすべてのデータベースとテーブルが自動的に共有されます。個々のオブジェクトの共有を設定する必要はありません。

データカタログでは、名前空間またはクラスターごとにフェデレーティッドカタログを作成できます。データカタログ外のエンティティを指す場合、カタログはフェデレーティッドカタログと呼ばれます。Amazon Redshift 名前空間のテーブルとビューは、データカタログに個別のテーブルとして表示されます。フェデレーティッドデータベースは、同じアカウント内または Lake Formation の別のアカウント内の、選択した IAM プリンシパルおよび SAML ユーザーと共有できます。行と列のフィルター式を含めて、特定データへのアクセスを制限することもできます。詳細については、「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」を参照してください。

データカタログは、カタログ、データベース、テーブル (およびビュー) で構成される 3 レベルのメタデータ階層をサポートします。名前空間をデータカタログに登録すると、Amazon Redshift データ階層は次のようにデータカタログの 3 レベルの階層にマッピングされます。

- Amazon Redshift 名前空間は、データカタログのマルチレベルカタログになります。
- 関連付けられた Amazon Redshift データベースは、データカタログにカタログとして登録されます。
- Amazon Redshift スキーマは、データカタログ内のデータベースになります。
- Amazon Redshift テーブルは、データカタログ内のテーブルになります。



この 3 レベルのメタデータ階層では、データカタログの「catalog1/catalog2.database.table」という 3 つの部分からなる表記を使用して Amazon Redshift テーブルにアクセスできます。また、データチームは、Amazon Redshift が データカタログアカウント内のテーブルを整理するために使用するのと同じ組織を維持できます。

Lake Formation では、データカタログリソースのきめ細かなアクセスコントロールを使用して、Amazon Redshift からのデータを安全に管理できます。この統合により、共通のアクセスコントロールメカニズムを使用して、単一のカタログから分析データを管理、保護、クエリできます。

制限事項については、「Amazon Redshift データウェアハウスデータを に取り込む際の制限 AWS Glue Data Catalog」を参照してください。

トピック

- 主な利点
- 役割と責任
- で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog
- Amazon Redshift フェデレーティッドカタログの作成
- カタログオブジェクトの表示
- フェデレーティッドカタログの更新
- 共有フェデレーティッドカタログへのアクセス

- フェデレーティッドカタログの削除
- フェデレーティッドカタログのクエリ
- その他のリソース

主な利点

Amazon Redshift クラスターと名前空間を に登録 AWS Glue Data Catalog し、Amazon S3 データレイクと Amazon Redshift データウェアハウス間でデータを統一すると、次の利点があります。

- 統一されたクエリエクスペリエンス – データを移動したりコピーしたりせずに、Amazon EMR Serverless や Amazon Athena など、Apache Iceberg と互換性のある任意のクエリエンジンを使用して、Amazon Redshift マネージドデータとAmazon S3 バケット内のデータをクエリします。
- サービス間の一貫したデータアクセス – データソースが Data Catalog に登録されているため、異なる AWS 分析サービスから同じフェデレーティッドデータソースにアクセスするときに、データパイプラインのデータベース名とテーブル名を更新する必要はありません。
- きめ細かなアクセスコントロール – Lake Formation のアクセス許可を適用して、きめ細かなアクセスコントロールのアクセス許可を使用してフェデレーティッドデータソースへのアクセスを管理できます。

役割と責任

ロール	責任
Amazon Redshift プロデューサークラスター管理者	クラスターまたは名前空間をデータカタログに登録します。
Lake Formation データレイク管理者	クラスターまたは名前空間の招待を受け入れ、フェデレーティッドカタログを作成し、フェデレーティッドカタログへのアクセスを他のプリンシパルに付与します。
Lake Formation 読み取り専用管理者	フェデレーティッドカタログを検出し、フェデレーティッドカタログ内の Amazon Redshift テーブルをクエリします。

データ転送ロール	Amazon Redshift は、ユーザーに代わって Amazon S3 バケットとの間でデータを転送します。

以下は、Amazon Redshift 名前空間へのアクセスをユーザーに許可するための大まかな手順です。

1. Amazon Redshift では、プロデューサークラスター管理者はクラスターまたは名前空間をデータカタログに登録します。
2. データレイク管理者は、Amazon Redshift プロデューサークラスター管理者からの名前空間の招待を受け入れ、データカタログにフェデレーティッドカタログを作成します。

   この手順を完了すると、データカタログ内で Amazon Redshift 名前空間カタログを管理できます。
3. カタログ、データベース、テーブルに対するアクセス許可をユーザーに付与します。名前空間カタログ全体またはテーブルのサブセットを、同じアカウント内または別のアカウント内のユーザーと共有できます。

で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog

1. データレイク管理者の作成 - 名前空間の招待を受け入れる権限を持つ IAM ロールを作成し、 AWS Glue Data Catalog オブジェクト (カタログ、データベース、テーブル/ビュー) を作成し、他のユーザーに Lake Formation アクセス許可を付与します。

   データレイク管理者を作成するための詳しい手順については、「データレイク管理者を作成する」を参照してください。
2. データレイク管理者アクセス許可を更新します。

   データレイク管理者のアクセス許可に加えて、データレイク管理者は、Lake Formation で Amazon Redshift 名前空間の招待を受け入れ、データカタログリソースを作成または更新し、データレイクアクセスを有効にするために、次のアクセス許可が必要です。

   JSON

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

```
    "Id": "glue-enable-datalake-access",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "redshift:AssociateDataShareConsumer",
                "redshift:DescribeDataSharesForConsumer",
                "redshift:DescribeDataShares",
                "redshift-serverless:CreateNamespace",
                "redshift-serverless:CreateWorkgroup",
                "redshift-serverless:DeleteNamespace",
                "redshift-serverless:DeleteWorkgroup",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeAvailabilityZones",
                "s3:createBucket",
                "s3:deleteBucket",
                "s3:putBucketPolicy",
                "s3:putEncryptionConfiguration",
                "s3:putLifecycleConfiguration",
                "s3:putBucketVersioning",
                "iam:CreateRole"
            ],
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/data transfer role name",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "glue.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

3. フェデレーティッドカタログの作成に使用される IAM ロールがデータレイク管理者でない場合は、ロールに `Create catalog` アクセス許可を付与する必要があります。

   カタログ作成者を作成するには

   a. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   b. ナビゲーションペインの [管理] で、[管理ロールとタスク] を選択します。

   c. [付与] を選択します。

   d. [アクセス許可の付与] 画面で、IAM ユーザーとロールを選択します。

   e. [カタログ作成] のアクセス許可を選択します。

   f. 必要に応じて、付与可能な [カタログ作成] アクセス許可を付与することもできます。付与可能なアクセス許可により、カタログ作成者は他のプリンシパルに `Create catalog` アクセス許可を付与できます。

   g. [付与] を選択します。

   AWS CLI フェデレーティッドカタログを作成するアクセス許可を付与する の例。

```
aws lakeformation grant-permissions \
--cli-input-json \
'{
    "Principal": {
     "DataLakePrincipalIdentifier":"arn:aws:iam::123456789012:role/Admin"
    },
    "Resource": {
        "Catalog": {
        }
    },
    "Permissions": [
        "CREATE_CATALOG",
        "DESCRIBE"
    ]
}'
```

4. 読み取り専用管理者ロールを作成して、Amazon Redshift Query Editor V2 からデータカタログ内の Amazon Redshift フェデレーティッドカタログを検出します。

   Amazon Redshift Query Editor V2 からフェデレーティッドカタログ内の Amazon Redshift テーブルをクエリするには、読み取り専用管理者ロールポリシーに、Amazon Redshift サービスにリンクされたロール `AWSServiceRoleForRedshift` の ARN が含まれていることを確認します。

```
 aws lakeformation put-data-lake-settings
        --region us-east-1 \
        --data-lake-settings \
 '{
   "DataLakeAdmins": [{"DataLakePrincipalIdentifier":"arn:aws:iam::123456789012:role/
Admin"}],
   "ReadOnlyAdmins":[{"DataLakePrincipalIdentifier":"arn:aws:iam::123456789012:role/
aws-service-role/redshift.amazonaws.com/AWSServiceRoleForRedshift"}],
   "CreateDatabaseDefaultPermissions":[],
   "CreateTableDefaultPermissions":[],
   "Parameters":{"CROSS_ACCOUNT_VERSION":"4","SET_CONTEXT":"TRUE"}
  }'
```

5. ユーザーに代わって Amazon S3 バケットとの間でデータを転送するために Amazon Redshift が引き受けることができるデータ転送ロールを作成します。

   Athena、Amazon EC2 での Amazon EMR などの Apache Iceberg 互換クエリエンジンのデータレイクアクセスを有効にしてデータカタログ内の Amazon Redshift リソースにアクセスする場合は、Amazon S3 バケットとの間でデータ転送を実行するために必要なアクセス許可を持つ IAM ロールを作成する必要があります。

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Id": "glue-enable-datalake-access",
       "Statement": [{
         "Sid": "DataTransferRolePolicy",
           "Effect": "Allow",
           "Action": [ "glue:GetCatalog",
                       "glue:GetDatabase",
                       "kms:GenerateDataKey",
                       "kms:Decrypt"],
           "Resource": "*"
       }
       ]
   }
   ```

6. AWS Glue および Amazon Redshift サービスのデータ転送ロールに次の信頼ポリシーを追加して、Amazon S3 バケットとの間でデータを転送するロールを引き受けます。

JSON

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

7. カスタマーマネージドキーを使用して Amazon Redshift クラスター/名前空間内のデータを暗号化 AWS KMS する場合は、次のキーポリシーをキーに追加します。アカウント番号を有効な AWS アカウント番号に置き換え、データ転送ロール名を指定します。デフォルトでは、Amazon Redshift クラスターのデータは、KMS キーを使用して暗号化されます。Lake Formation には、暗号化用のカスタム KMS キーを作成するオプションがあります。カスタマーマネージドキーを使用している場合は、特定のキーポリシーをキーに追加する必要があります。

   カスタマーマネージドキーの許可を管理する方法の詳細については、カスタマーマネージドキーを参照してください。

   JSON

```
{
    "Version":"2012-10-17",
    "Id": "auto-redshift-3",
    "Statement": [{
            "Sid": "RedshiftAllowAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
```

```
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:CallerAccount": "111122223333",
                    "kms:ViaService": "redshift.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "Sid": "RedshiftServerlessAllowAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:CallerAccount": "111122223333",
                    "kms:ViaService": "redshift-serverless.us-
east-1.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DirectMetadataAccess",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:Describe*",
```

```
                "kms:Get*",
                "kms:List*",
                "kms:RevokeGrant"
            ],
            "Resource": "*"
        },
        {
            "Sid": "GenerateDataKeyDecryptDataTransferRole",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/data-transfer-role-name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                "kms:ViaService": "s3.us-east-1.amazonaws.com"
                }
            }
        }
    ]
}
```

Amazon Redshift フェデレーティッドカタログの作成

このトピックでは、クラスターまたは名前空間の招待を受け入れ、フェデレーティッドマルチレベルカタログを作成し、他のプリンシパルにアクセス許可を付与するために必要な手順について説明します。これらのタスクは、Lake Formation コンソール、 AWS Command Line Interface (AWS CLI)、または APIs/SDKs。このトピックの例では、同じアカウント内のプロデューサークラスター/名前空間、データカタログ、およびデータコンシューマーを示しています。

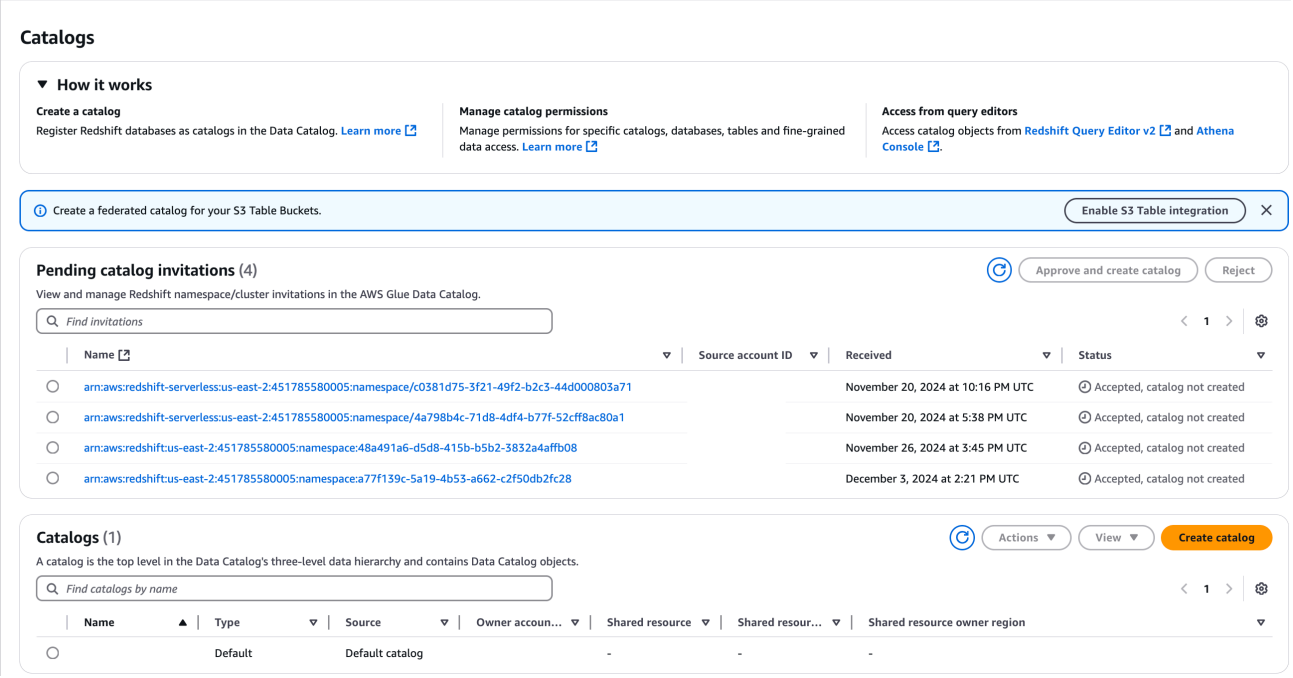
Lake Formation のクロスアカウント機能の詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

データカタログで Amazon Redshift 名前空間を管理するには

1. 名前空間への招待を確認して承諾します。

   Console

   1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にデータレイク管理者としてサインインします。[データカタログ] の [カタログ] ページに移動します。
   2. アクセスが許可されている名前空間の招待を確認します。[ステータス] 列は、名前空間の現在の参加ステータスを示します。[未承諾] ステータスは、名前空間に追加されたものの、まだ承諾していないか招待を拒否したことを示します。



   3. 名前空間またはクラスターの招待に応答するには、招待名を選択し、[招待の確認] を選択します。[招待の承諾または拒否] で、招待の詳細を確認します。[Accept] (承諾) を選択して招待を承諾するか、[Reject] (拒否) を選択して招待を却下します。招待を拒否した場合、名前空間にはアクセスできません。

   AWS CLI

   以下の例では、招待を表示、承諾、登録する方法を示します。 AWS アカウント ID を有効な AWS アカウント ID に置き換えます。`data-share-arn` を、名前空間を参照する実際の Amazon リソースネーム (ARN) に置き換えます。

1. 保留中の招待を確認します。

```
aws redshift describe-data-shares \
 --data-share-arn 'arn:aws:redshift:us-
east-1:123456789012:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/
ds_internal_namespace' \
```

2. 招待を承諾します。

```
 aws redshift associate-data-share-consumer \
 --data-share-arn 'arn:aws:redshift:us-
east-1:123456789012:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/
ds_internal_namespace' \
 --consumer-arn 'arn:aws:glue:us-east-1:123456789012:catalog'
```

3. Lake Formation アカウントにクラスターまたは名前空間を登録します。RegisterResource API オペレーションを使用して、データ共有を Lake Formation に登録します。DataShareArn は ResourceArn の入力パラメータです。

> **Note**
>
> これは必須の手順です。

```
aws lakeformation register-resource \
 --resource-arn 'arn:aws:redshift:us-
east-1:123456789012:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/
ds_internal_namespace'
```

2. フェデレーティッドカタログを作成します。

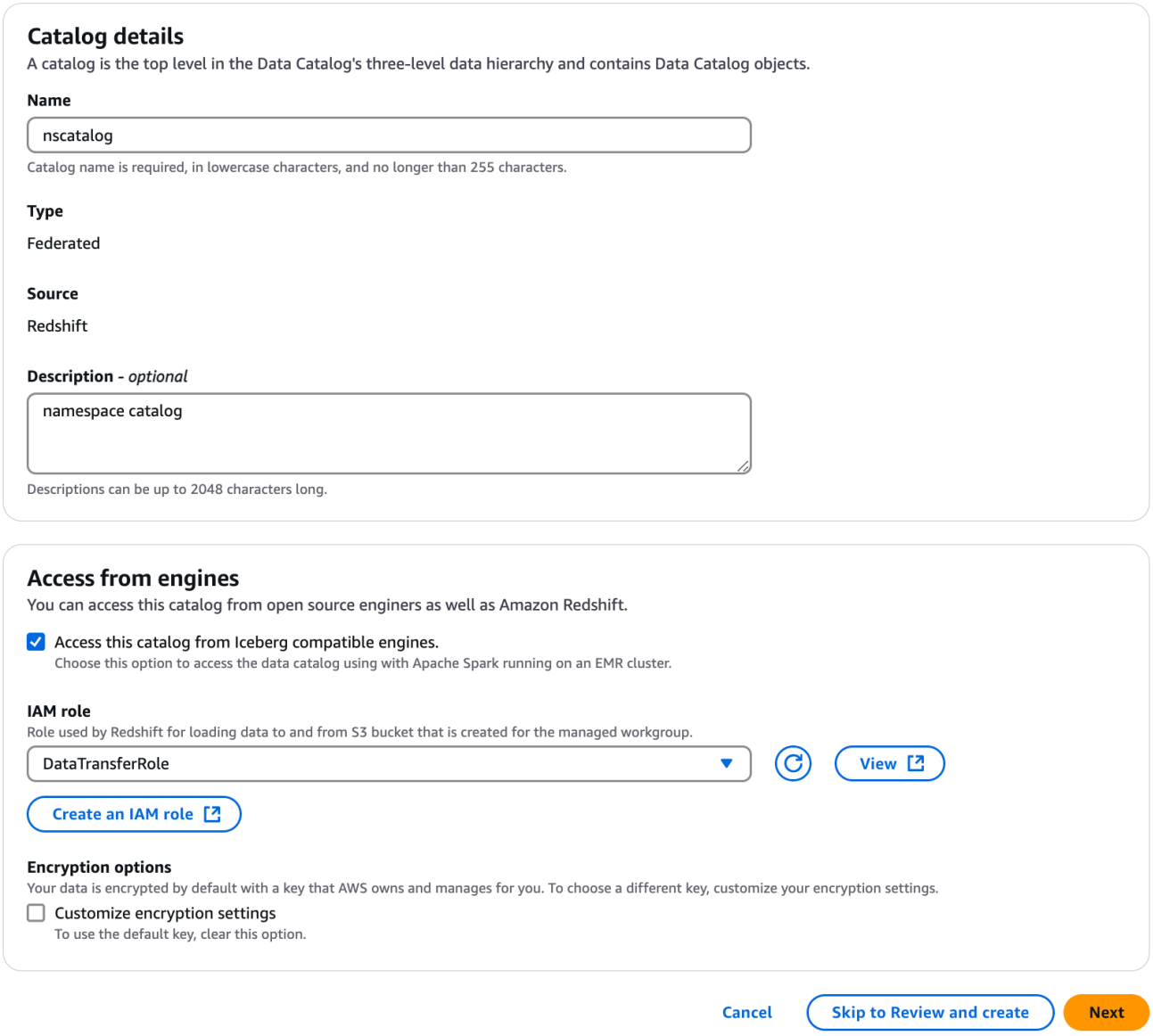
   招待を承諾したら、Amazon Redshift 名前空間内のオブジェクトをデータカタログにマッピングするフェデレーティッドカタログをデータカタログに作成する必要があります。データレイク管理者、またはカタログの作成に必要なアクセス許可を持つユーザーもしくはロールである必要があります。

   Console

   1. 名前空間の[招待]を受け入れると、[カタログの詳細の設定] ページが表示されます。

2. [カタログの詳細の設定] ページで、カタログの一意の名前を入力します。カタログ名には小文字を使用します。カタログ名は 255 文字以下にする必要があります。この識別子を使用して、名前空間の内部をメタデータ階層 (catalogid.dbName.schema.table) にマッピングします。
3. キーの説明を入力します。説明は 2,048 文字以下にする必要があります。
4. 次に、[Iceberg 互換エンジンからこのカタログにアクセスする] チェックボックスにチェックを入れると、Athena や Amazon EMR での Apache Spark のような Apache Iceberg 互換の分析エンジンを使用して、Amazon Redshift リソースにアクセスできるようになります。

   Amazon Redshift を使用してフェデレーティッドカタログにアクセスするためにデータレイクのアクセスを有効にする必要はありません。



5. これらのクエリエンジンが Amazon Redshift 名前空間に読み書きできるようにするには、Amazon Redshift データウェアハウスのワークロードに影響を与えることなく、読み取りおよび書き込みオペレーションを実行するために必要なコンピューティングリソースとストレージリソースを備えたマネージド Amazon Redshift クラスター AWS Glue を作成します。

   Amazon S3 バケットとの間でデータを転送するために必要なアクセス許可を持つ IAM ロールも指定する必要があります。

6. デフォルトでは、Amazon Redshift クラスター内のデータは AWS マネージドキーを使用して暗号化されます。Lake Formation には、暗号化用のカスタム KMS キーを作成するオプションがあります。カスタマーマネージドキーを使用している場合は、特定のキーポリシーをキーに追加する必要があります。

   カスタマーマネージドキーを使用して Amazon Redshift クラスター/名前空間のデータを暗号化している場合は、[暗号化設定のカスタマイズ] を選択します。カスタムキーを使用するには、KMS キーにカスタムマネージドキーポリシーを追加する必要があります。詳細については、「で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog」を参照してください。

AWS CLI

次のサンプルコードを使用し、 AWS CLIを使用してデータカタログに公開された Amazon Redshift データを含むカタログを作成します。

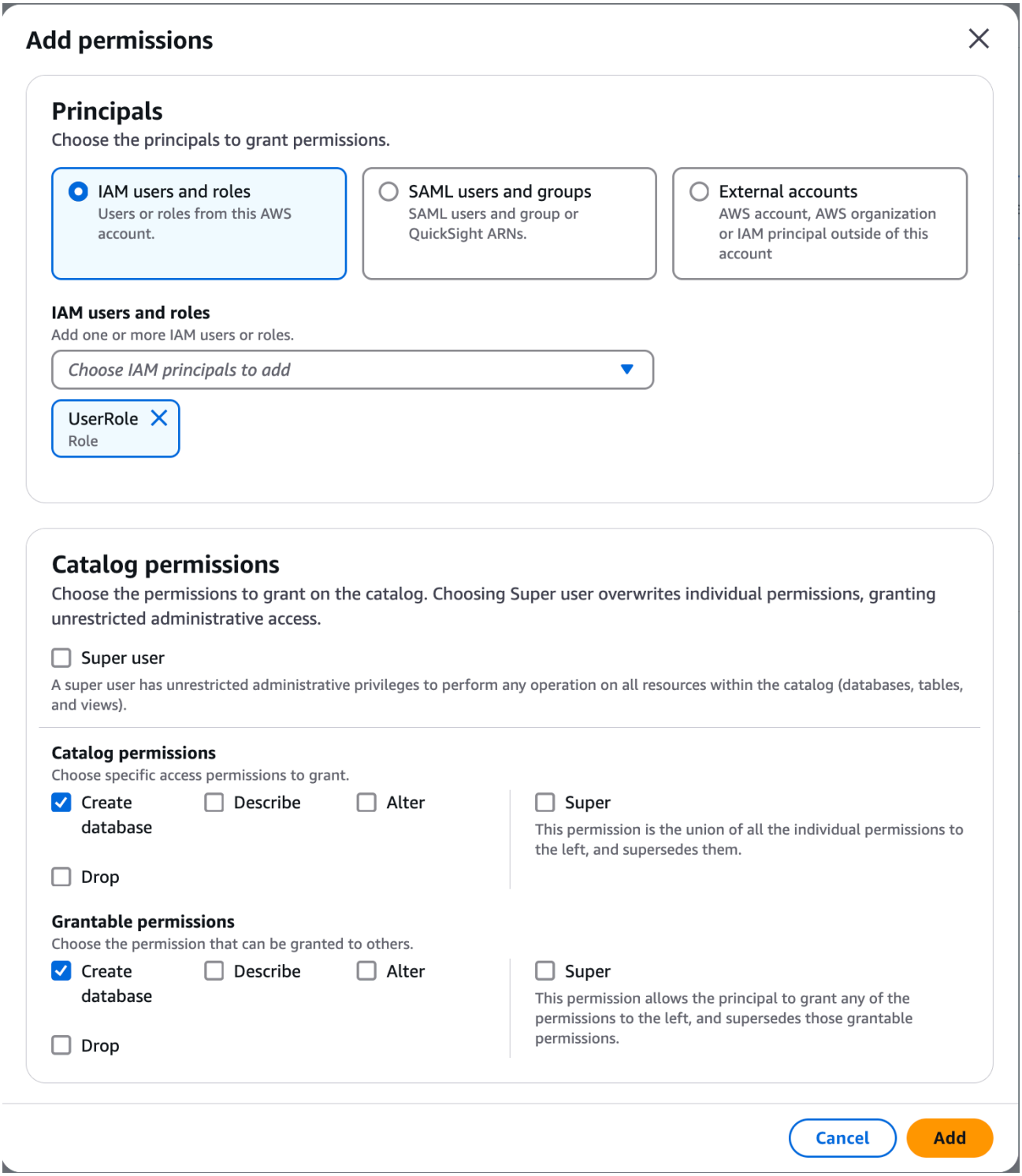
```
aws glue create-catalog
--cli-input-json \
'{
    "Name": "nscatalog",
    "CatalogInput": {
        "Description": "Redshift federated catalog",
        "CreateDatabaseDefaultPermissions" : [],
        "CreateTableDefaultPermissions": [],
        "FederatedCatalog": {
            "Identifier": "arn:aws:redshift:us-
east-1:123456789012:datashare:11524d7f-f56d-45fe-83f7-d7bb0a4d6d71/
ds_internal_namespace",
            "ConnectionName": "aws:redshift"
        },
        "CatalogProperties": {
```

```
          "DataLakeAccessProperties" : {
            "DataLakeAccess" : true,
            "DataTransferRole" :
 "arn:aws:iam::123456789012:role/DataTransferRole"
         }
       }
    }
}'
```

3. 自分のアカウントまたは外部アカウントのユーザーに許可を付与します。

   AWS マネジメントコンソール

   1. 共有のカタログ、データベース、テーブルに対するアクセス許可を他のユーザーに付与するには、[次へ] を選択します。
   2. [アクセス許可を追加] 画面で、プリンシパルと、付与するアクセス許可のタイプを選択します。



a. [Principals] (プリンシパル) セクションでプリンシパルタイプを選択してから、アクセス許可の付与先となるプリンシパルを指定します。

- IAM ユーザーとロール – IAM ユーザーとロールのリストから、ユーザーまたはロールを 1 つ以上選択します。

- SAML ユーザーとグループ – SAML および Amazon Quick ユーザーとグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、または Amazon Quick ユーザーまたはグループの ARNs を入力します。各 ARN の後で [Enter] キーを押します。

  ARNs」を参照してください。 AWS CLI AWS CLI

- 外部アカウント – AWS、 AWS 組織、または IAM プリンシパルには、IAM ユーザーまたはロールの 1 つ以上の有効な AWS アカウント IDs、組織 IDs、組織単位 IDs、または ARN を入力します。各 ID の後で [Enter] キーを押します。組織 ID は、最初の「o-」と、その後に続く 10 ~ 32 個の小文字または数字で構成されています。組織単位 ID は「ou-」で始まり、その後に 4 ~ 32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」ダッシュと 8 ~ 32 個の追加の小文字または数字が続きます。

b. [Permissions] (許可) セクションで、許可と付与可能な許可を選択します。

   [カタログのアクセス許可] で、付与するアクセス許可を 1 つ以上選択します。「付与可能なアクセス許可」で、アカウントの他のプリンシパルに付与できるアクセス許可を選択します AWS 。このオプションは、外部アカウントから IAM プリンシパルにアクセス許可を付与する場合はサポートされません。

   [スーパーユーザー] を選択して、カタログ内のリソース (データベース、テーブル、ビュー) に対する無制限のアクセス許可をユーザーに付与します。

3. [Add] (追加) を選択します。

AWS CLI

以下の例を使用して、 AWS CLIでカタログ、データベース、テーブルのアクセス許可を付与します。

- 次の例は、フェデレーティッドカタログに対するアクセス許可を付与する方法を示しています。

```
aws lakeformation grant-permissions
 --cli-input-cli-json \
   '{
         "Principal": {
              "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:role/
non-admin"
```

```
          },
          "Resource": {
               "Catalog": {
                     "Id": "123456789012:nscatalog"
                }
            },
            "Permissions": [
                   "DESCRIBE","CREATE_CATALOG"
             ],
            "PermissionsWithGrantOption": [
             ]
    }'
```

- 次の例を使用して、データベースに対するアクセス許可を付与します。

```
aws lakeformation grant-permissions \
  --cli-input-json \
          '{
              "Principal": {

 "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:role/non-admin"
              },
              "Resource": {
                  "Database": {
                      "CatalogId": "123456789012:nscatalog/dev",
                      "Name": "public"
                  }
              },
              "Permissions": [
                  "ALL"
              ]
          }'
```

- 次の例は、Amazon Redshift データベースのテーブルに対するアクセス許可を付与する方法を示しています。

```
aws lakeformation grant-permissions \
  --cli-input-json \
        '{
            "Principal": {

 "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:role/non-admin"
            },
```

```
            "Resource": {
                "Table": {
                    "CatalogId": "123456789012:nscatalog2/dev",
                    "DatabaseName": "public",
                    "TableWildcard" : {}
                }
            },
            "Permissions": [
                "ALL"
            ]
        }'
```

4. [次へ] を選択してカタログの詳細を確認し、フェデレーティッドカタログを作成します。新しく作成されたフェデレーティッドカタログとカタログオブジェクトが[カタログ] ページに表示されます。

   Amazon Redshift フェデレーティッドカタログを参照するには `catalogID = 123456789012:Redshift-federated catalog id` を使用します。

カタログオブジェクトの表示

フェデレーティッドカタログを作成したら、Lake Formation コンソールまたは AWS CLIを使用してカタログ内のオブジェクトを表示できます。

AWS マネジメントコンソール

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. [データカタログ] で [カタログ] を選択します。
3. [カタログ] ページのリストからフェデレーティッドカタログを選択します。
4. カタログの概要ページには、アクセス許可があるカタログオブジェクト (データベースとテーブル) が表示されます。[アクセス許可] タブには、これらのオブジェクトに対するアクセス許可が付与された IAM プリンシパルが表示されます。

AWS CLI

- 次の AWS CLI 例は、最上位カタログをリクエストする方法を示しています。

```
aws glue get-catalog \
--catalog-id 123456789012:nscatalog
```

[応答]

```
{
    "Catalog": {
        "CatalogId": "123456789012:nscatalog",
        "Name": "nscatalog",
        "ResourceArn": "arn:aws:glue:us-east-1:123456789012:catalog/nscatalog",
        "Description": "Redshift published Catalog",
        "CreateTime": "2024-09-05T14:49:16-07:00",
        "FederatedCatalog": {
            "Identifier": "arn:aws:redshift:us-
east-1:123456789012:datashare:b1234589-e823-4a14-ad8e-077085540a50/
ds_internal_namespace",
            "ConnectionName": "aws:redshift"
        },
        "CatalogProperties": {
            "DataLakeAccessProperties": {
                "DataLakeAccess": true,
                "DataTransferRole": "arn:aws:iam::123456789012:role/
DataTransferRole",
                "KmsKey": "AWS_OWNED_KMS_KEY",
                "ManagedWorkgroupName": "123456789012:nscatalog",
                "ManagedWorkgroupStatus": "AVAILABLE",
                "RedshiftDatabaseName": "dev"
            }
        },
        "CatalogIdentifier": "e2309c2c2fb048f1a3069dfdc1c7883e",
        "CreateTableDefaultPermissions": [],
        "CreateDatabaseDefaultPermissions": []
    }
}
```

- 次の例は、アカウント内のすべてのカタログをリクエストする方法を示しています。

```
aws glue get-catalogs \
  --recursive
```

- 次のリクエスト例は、Amazon Redshift データベースレベルのカタログを取得する方法を示しています。

```
aws glue get-catlog \
```

```
 --catalog-id 123456789012:namespace catalog name/redshift database name
```

- 次のリクエスト例は、Amazon Redshift データベースレベルのカタログでデータベースを取得する方法を示しています。

```
aws glue get-databases \
--catalog-id 123456789012:namespace catalog name/redshift database name
```

- 次のリクエスト例は、カタログで Amazon Redshift テーブルを取得する方法を示しています。

```
aws glue get-table \
  --catalog-id 123456789012:parent catalog name/redshift database \
  --database-name redshift schema name \
  --name table name
```

- 次の例は、Amazon Redshift データベースからテーブルを取得する方法を示しています。

```
aws glue get-tables \
 --catalog-id 123456789012:namespace catalog name/redshift database name \
 --database-name RS schema name
```

フェデレーティッドカタログの更新

Lake Formation コンソール、、 AWS CLI または UpdateCatalog API オペレーションを使用して、データカタログで Amazon Redshift フェデレーティッドカタログを更新できます。

AWS マネジメントコンソール

Lake Formation コンソールを使用してフェデレーティッドカタログを更新するには、次の手順に従います。

1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールを開きます。
2. 左側のナビゲーションペインの [データカタログ] で、[カタログ] を選択します。
3. [カタログ] ページで、更新する Amazon Redshift フェデレーティッドカタログを選択します。
4. [アクション] で [編集] を選択します。

5. [カタログの詳細の設定] 画面の [エンジンからアクセス] セクションで、[Iceberg 互換エンジンからこのカタログにアクセスする] を選択します。このオプションにチェックを入れると、Apache Iceberg 互換クエリエンジンのデータレイクアクセスが有効になります。
6. 次に、新しい IAM ロールを作成するか、Amazon S3 バケットとの間でデータ転送を実行するアクセス許可を付与するポリシーを持つ既存の IAM ロールを選択します。

   アクセス許可の詳細については、「で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog」を参照してください。
7. デフォルトでは、Amazon Redshift クラスターのデータは、 AWS マネージドキーを使用して暗号化されます。カスタマーマネージドキーを使用してデータを暗号化する場合は、KMS キーを作成するか、で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog セクションで定義されたアクセス許可を持つ既存のキーを選択します。
8. [保存] を選択します。

   正常に完了すると、[カタログの詳細] ページに、ステータスが「成功」であるマネージドワークグループ名が表示されます。

AWS CLI

DataLakeAacess パラメータ値を false に設定して、データレイクアクセスを無効にした update-catalog CLI 入力の例を次に示します。

```
aws glue update-catalog  --cli-input-json \
'{
    "Name": "nscatalog",
    "CatalogInput": {
        "Description": "Redshift published catalog",
        "CreateDatabaseDefaultPermissions" : [],
        "CreateTableDefaultPermissions": [],
        "FederatedCatalog": {
            "Identifier": "arn:aws:redshift:us-
east-1:123456789012:datashare:11524d7f-f56d-45fe-83f7-d7bb0a4d6d71/
ds_internal_namespace",
            "ConnectionName": "aws:redshift"
        },
        "CatalogProperties": {
          "DataLakeAccessProperties" : {
            "DataLakeAccess" : false
        }
       }
```

```
    }
}'
```

共有フェデレーティッドカタログへのアクセス

AWS Lake Formation クロスアカウント機能を使用すると、ユーザーは分散データレイクを複数の AWS アカウント、 AWS 組織間で安全に共有したり、別のアカウントの IAM プリンシパルと直接共有したりして、メタデータや基盤となるデータにきめ細かなアクセスを提供したりできます。

Lake Formation は AWS Resource Access Manager 、 (AWS RAM) サービスを使用してリソース共有を容易にします。カタログリソースを別のアカウントと共有すると、 は、リソース許可を承認または拒否するための招待を被付与者アカウント AWS RAM に送信します。

Amazon Athena や Redshift Spectrum などの統合されたサービスでは、クエリに共有リソースを含めることができるように、リソースリンクが必要になります。プリンシパルは、別の から共有リソース AWS Glue Data Catalog へのリソースリンクを に作成する必要があります AWS アカウント。リソースリンクの詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。

カタログリンクコンテナは、他のアカウントのローカルまたはクロスアカウントのフェデレーションデータベースレベルのカタログを参照する Data Catalog オブジェクトです AWS 。カタログリンクコンテナ内にデータベースリンクとテーブルリンクを作成することもできます。データベースリンクまたはテーブルリンクを作成するときは、同じターゲット Amazon Redshift データベースレベルカタログ (Amazon Redshift データベース) に存在するターゲットリソースを指定する必要があります。

カタログリンクコンテナを作成するには、Lake Formation の `CREATE_CATALOG` または `glue:CreateCatalog` アクセス許可が必要です。

フェデレーションカタログ内のデータベースまたはテーブルを 間で共有するには、クロスアカウントバージョン設定バージョン 4 AWS アカウント以降が必要です。

クロスアカウントフェデレーティッドカタログへのカタログリンクコンテナの作成

AWS Lake Formation コンソール、`CreateCatalog`API、 AWS Glue または AWS Command Line Interface () を使用して、任意の AWS リージョンの Redshift データベースレベルのフェデレーティッドカタログを指すカタログリンクコンテナを作成できますAWS CLI。

共有カタログへのカタログリンクコンテナを作成するには (コンソール)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。Lake Formation CREATE_CATALOG アクセス許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインで [カタログ] を選択してから、[カタログの作成] を選択します。
3. [カタログの詳細の設定] ページで、以下の情報を指定します。

   名前

   カタログ名と同じルールに従う名前を入力します。名前は、ターゲット共有カタログと同じものにすることができます。

   タイプ

   カタログのタイプとして [カタログリンクリンクコンテナ] を選択します。

   ソース

   Redshift を選択してください。

   ターゲット Redshift カタログ

   Redshift のデータベースレベルのフェデレーティッドカタログを選択するか、リストからローカル (所有) カタログを選択します。

   このリストには、アカウントと共有されているすべてのカタログが含まれています。各カタログにカタログ所有者アカウント ID が記載されていることに注意してください。アカウントと共有されていることが分かっているカタログが表示されない場合は、以下を確認してください。

   - データレイク管理者でない場合は、データレイク管理者からそのカタログに対する Lake Formation アクセス許可が付与されていることを確認します。
   - データレイク管理者であり、アカウントが付与元のアカウントと同じ AWS 組織にない場合は、カタログに関する AWS Resource Access Manager (AWS RAM) リソース共有招待を承諾していることを確認します。詳細については、「からのリソース共有の招待の承諾 AWS RAM」を参照してください。

   > **Note**
   >
   > コンソールを使用してカタログリンクコンテナを作成する場合、クロスアカウント Redshift カタログを選択しようとすると、[ターゲット Redshift カタログ]

ドロップダウンが No matches を表示することがあります。この表示にもかかわらず、Amazon Redshift フェデレーティッドデータベースレベルカタログ (Amazon Redshift データベース) のターゲット ARN を入力フィールドに手動で入力しても、フォームは引き続き正常に動作します。例: arn:aws:glue:us-east-1:123456789012:catalog/federated-catalog-redshift/dev。この動作は、コンソールが現在サインインしているアカウント内のみで潜在的な候補を検索できるために発生します。ドロップダウンは自動入力機能を目的としていますが、クロスアカウントアクセスのために ARN を手動で入力することもできます。

4. Apache Iceberg クエリエンジンが Amazon Redshift 名前空間を読み書きできるようにするには、Amazon Redshift データウェアハウスのワークロードに影響を与えることなく、読み取りおよび書き込みオペレーションの実行に必要なコンピューティングリソースとストレージリソースを備えたマネージド Amazon Redshift クラスター AWS Glue を作成します。Amazon S3 バケットとの間でデータを転送するために必要なアクセス許可を持つ IAM ロールを指定する必要があります。
5. [次へ] を選択します。
6. (オプション) [アクセス許可を追加] を選択して、他のプリンシパルにアクセス許可を付与します。

   ただし、カタログリンクコンテナに対してアクセス許可を付与しても、ターゲットの (リンクされた) カタログへのアクセス許可は付与されません。Athena でカタログリンクが表示されるようにするには、ターゲットのカタログに対するアクセス許可を別途付与する必要があります。
7. 次に、カタログリンクコンテナの詳細を確認し、[カタログを作成] を選択します。

   これにより、[カタログ] ページでリンクコンテナ名を表示できます。

   これで、カタログリンクコンテナにデータベースリンクとテーブルリンクを作成して、クエリエンジンからのアクセスを有効にすることができます。

カタログリンクコンテナ CLI の作成例

- 次の例では、TargetRedshiftCatalog オブジェクトは Amazon Redshift フェデレーティッドデータベースレベルカタログ (Amazon Redshift データベース) の ARN を指定します。カタログリンクコンテナを作成する際は、DataLakeAccess を有効にする必要があります。

```
aws glue create-catalog \
  --cli-input-json
```

```
    '{
        "Name": "linkcontainer",
        "CatalogInput": {
            "TargetRedshiftCatalog": {
               "CatalogArn": "arn:aws:us-east-1:123456789012:catalog/nscatalog/dev"
             },
            "CatalogProperties": {
              "DataLakeAccessProperties" : {
                "DataLakeAccess" : true,
                "DataTransferRole" : "arn:aws:iam::111122223333:role/
DataTransferRole"
             }
           }
        }
    }'
```

カタログリンクコンテナでのリソースリンクの作成

データベースリンクとテーブルリンクへのリソースリンクは、カタログリンクコンテナの下に作成できます。データベースリソースリンクまたはテーブルリソースリンクを作成するときは、リンクコンテナが指すのと同じターゲット Amazon Redshift データベースレベルカタログ (Amazon Redshift データベース) に存在するターゲットリソースを指定する必要があります。

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、共有 Amazon Redshift データベースまたはテーブルへのリソースリンクを作成できますAWS CLI。

- 詳細な手順については、「共有 Data Catalog データベースへのリソースリンクの作成」を参照してください。

  以下は、カタログリンクコンテナの下にデータベースリソースリンクを作成する AWS CLI 例です。

```
aws glue create-database \
  --cli-input-json \
    '{
        "CatalogId": "111122223333:linkcontainer",
        "DatabaseInput": {
            "Name": "dblink",
             "TargetDatabase": {
               "CatalogId": "123456789012:nscatalog/dev",
```

```
                "DatabaseName": "schema1"
            }
        }
    }'
```

- カタログリンクコンテナの下にテーブルリソースリンクを作成するには、まずローカルに AWS Glue データベースを作成して AWS Glue Data Catalog 、テーブルリソースリンクを含める必要があります。

  共有テーブルへのリソースリンクの作成の詳細については、「共有 Data Catalog テーブルへのリソースリンクの作成」を参照してください。

  - テーブルリソースリンクを含むデータベースの作成例

```
aws glue create-database \
  --cli-input-json \
      '{
          "CatalogId": "111122223333:linkcontainer",
          "DatabaseInput": {
              "Name": "db1",
              "Description": "creating parent database for table link"
          }
      }'
```

  - テーブルリソースリンクの作成例

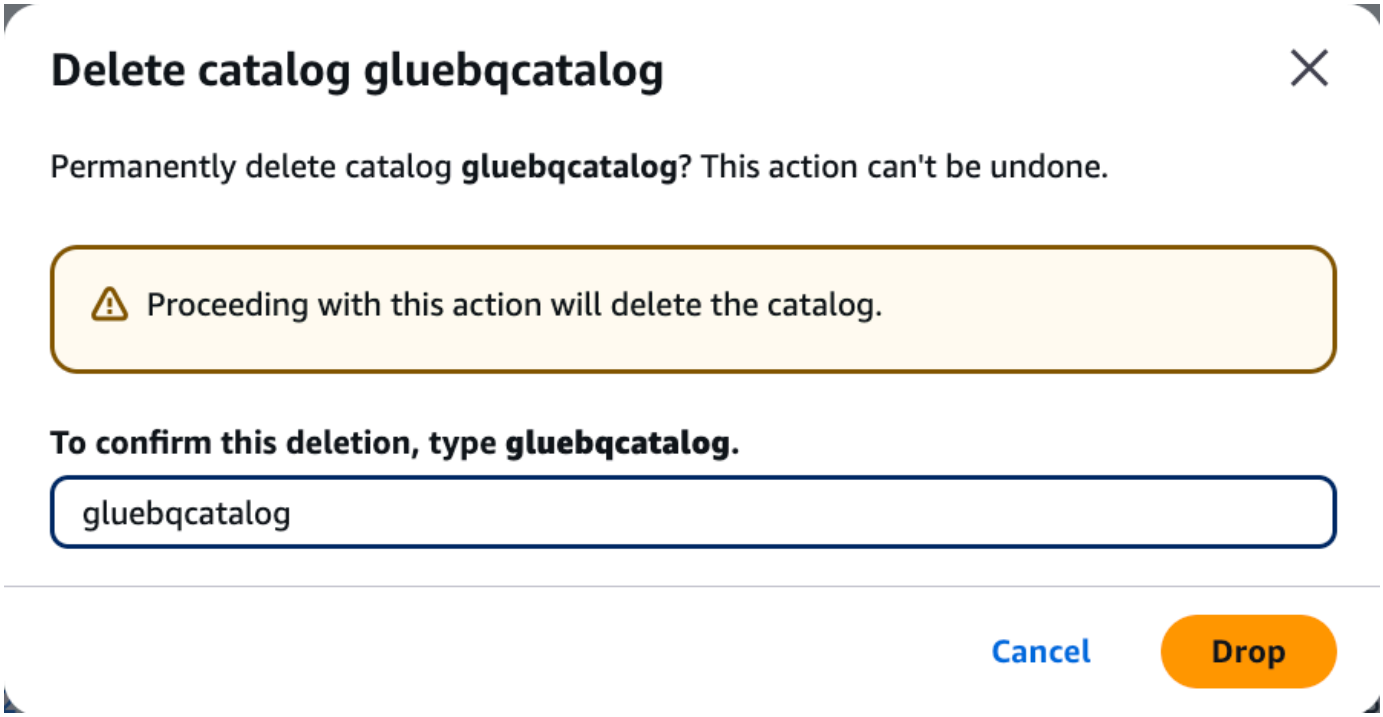
```
aws glue create-table \
  --cli-input-json \
    '{
        "CatalogId": "111122223333:linkcontainer",
         "DatabaseName": "db1",
        "TableInput": {
            "Name": "tablelink",
            "TargetTable": {
                "CatalogId": "123456789012:nscatalog/dev",
               "DatabaseName": "schema1",
                "Name": "table1"
             }
        }
    }'
```

フェデレーティッドカタログの削除

`glue:DeleteCatalog` オペレーションまたは AWS Lake Formation コンソール AWS Glue Data Catalog を使用して、 で作成したフェデレーティッドカタログを削除できます。

フェデレーティッドカタログを削除するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。
3. カタログリストから削除するカタログを選択します。
4. [アクション] で、[削除] を選択します。
5. [ドロップ] を選択して確認すると、データカタログからフェデレーティッドカタログが削除されています。



フェデレーティッドカタログを削除するには (CLI)

- ```
  aws glue delete-catalog
   --catalog-id 123456789012:catalog name
  ```
  ```

フェデレーティッドカタログのクエリ

他のプリンシパルにアクセス許可を付与した後、Amazon Redshift、Amazon EMR、、および AWS Glue ETL を使用して SQL ツールにログインすることで Amazon Athena、フェデレーティッドカタログのテーブルにサインインしてクエリを開始できます。

Apache Iceberg Rest 拡張エンドポイントまたはスタンドアロン Spark アプリケーション AWS Glue Data Catalog を使用して に接続する方法の詳細については、「 AWS Glue デベロッパーガイド」の「 セクションへのアクセス AWS Glue Data Catalog」を参照してください。

データ定義言語 (DDL) クエリを使用し、Amazon EMR の Apache Spark を使用してデータベース内でテーブルを作成および管理できます。Amazon Redshift データベースでテーブルを作成および削除するには、プリンシパルに Lake Formation `Create table`、`Drop` アクセス許可が必要です。

データカタログアクセス許可を付与する方法の詳細については、「データカタログリソースに対するアクセス許可の付与」を参照してください。

からのカタログリソースのクエリの詳細については Amazon Athena、Amazon Athena ユーザーガイド」の「 AWS Glue Data Catalog からのクエリ Amazon Athena」を参照してください。

その他のリソース

Amazon SageMaker Lakehouse を使用すると、データウェアハウスとデータレイクの両方でデータへの統合されたアクセスを実現できます。SageMaker Lakehouse を使用すると、オープンな Apache Iceberg REST API を通じて優先分析エンジン、機械学習エンジン、ビジネスインテリジェンスエンジンを使用し、一貫性のあるきめ細かなアクセスコントロールによってデータへの安全なアクセスを確保できます。

- Amazon SageMaker ワークショップ
- Amazon SageMaker Lakehouse を使用して企業のデータアクセスを簡素化する

の外部データソースへのフェデレーション AWS Glue Data Catalog

AWS Glue Data Catalog (データカタログ) は、Amazon Redshift、Snowflake、Amazon RDS などのクラウドデータベース Amazon DynamoDB、Oracle、Amazon MSK などのストリーミングサービス、および AWS Glue 接続を使用して Teradata などのオンプレミスシステムに接続できます。これ

らの接続は に保存 AWS Glue Data Catalog され、 に登録されるため AWS Lake Formation、使用可能なデータソースごとにフェデレーションカタログを作成できます。

フェデレーティッドカタログは、外部データシステムのデータベースを指す最上位コンテナです。これにより、抽出、変換、ロード (ETL) プロセスなしで、外部データシステムからデータを直接クエリできます。

AWS Glue 接続の詳細については、「 AWS Glue デベロッパーガイド」の「データへの接続」を参照してください。

データレイク管理者は、Amazon SageMaker Lakehouse または Amazon Athena を使用してフェデレーティッドカタログを作成できます。

データレイク管理者は、Lake Formation を使用してカタログ内のオブジェクトに対するきめ細かなアクセス許可を付与し、カタログ、データベース、テーブル、列、行、セルなどのさまざまなレベルでアクセスを制御できます。データアナリストは、Athena を使用してカタログ化されたデータソースを検出およびクエリし、Lake Formation は定義されたアクセスポリシーを適用します。アナリストは、各ソースに個別に接続することなく、1 つのクエリで複数のソース間のデータを結合できます。

トピック

- ワークフロー
- データカタログを外部データソースに接続するための前提条件
- AWS Glue 接続を使用したフェデレーティッドカタログの作成
- カタログオブジェクトの表示
- フェデレーティッドカタログの削除
- フェデレーティッドカタログのクエリ
- その他のリソース

ワークフロー

データレイク管理者または必要なアクセス許可を持つユーザーは、 AWS Glue Data Catalog を外部データソースに接続するための次のステップを完了します。

1. データソース AWS Glue への接続を作成します。接続を登録する場合、接続の登録に使用される IAM ロールは、Lambda 関数と Amazon S3 スピルバケットの場所にアクセスできる必要があります。

2. Lake Formation との接続を登録します。
3. 利用可能なデータソースに接続するための AWS Glue 接続を使用して、データカタログにフェデレーティッドカタログを作成します。データベース、テーブル、ビューはデータカタログ内で自動的にカタログ化され、Lake Formation に登録されます。
4. Lake Formation アクセス許可を使用して、特定のカタログ、データベース、およびテーブルへのアクセスをデータアナリストに許可します。Lake Formation を使用してデータレイク、ウェアハウス、OLTP ソース全体できめ細かなアクセスコントロールポリシーを定義できるため、行レベルと列レベルのセキュリティフィルターが可能になります。

   データアナリストは、個別の接続やデータソース認証情報を必要とせずに、Athena の SQL クエリを使用し、データカタログを介してすべてのデータにアクセスできるようになります。アナリストは、複数のソースからデータをスキャンするフェデレーティッド SQL クエリを実行し、複雑なデータパイプラインを使用せずにインプレースでデータを結合できます。

データカタログを外部データソースに接続するための前提条件

AWS Glue Data Catalog を外部データソースに接続し、接続を Lake Formation に登録して、フェデレーティッドカタログを設定するには、次の要件を満たす必要があります。

> **Note**
>
> Lake Formation データレイク管理者は、外部データソースに接続するための AWS Glue 接続を作成し、フェデレーティッドカタログを作成することをお勧めします。

1. IAM ロールを作成します。

   - 外部データソースへの接続を作成するために必要なリソース (Lambda 関数、Amazon S3 スピルバケット、IAM ロール、および AWS Glue 接続) をデプロイするために必要なアクセス許可を持つロールを作成します。
   - AWS Glue 接続プロパティ (Lambda 関数と Amazon S3 スピルバケット) にアクセスするために必要な最小限のアクセス許可を持つロールを作成します。これは、Lake Formation に接続を登録するときに含めるロールです。

     Lake Formation を使用してデータレイク内のデータを管理および保護するには、 AWS Glue 接続を Lake Formation に登録する必要があります。これにより、Lake Formation はフェデレーティッドデータソースをクエリするために Amazon Athena に認証情報を提供できます。

ロールには、Amazon S3 バケットと Lambda 関数に対する `Select` または `Describe` アクセス許可が必要です。

- s3:ListBucket
- s3:GetObject
- lambda:InvokeFunction

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1/object/*",
        "arn:aws:s3:::amzn-s3-demo-bucket1/object"
      ]
    },
    {
      "Sid": "lambdainvoke",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:123456789012:function:example-
lambda-function"
    },
    {
      "Sid": "gluepolicy",
      "Effect": "Allow",
      "Action": "glue:*",
      "Resource": "*"
    }
  ]
}
```

- 接続の登録に使用される IAM ロールに次の信頼ポリシーを追加します。

JSON

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

- 場所を登録するデータレイク管理者は、ロールに対する iam:PassRole アクセス許可を持っている必要があります。

  以下は、この許可を付与するインラインポリシーです。*<account-id>* を有効な AWS アカウント番号に置き換え、*<role-name>* をロールの名前に置き換えます。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PassRolePermissions",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/example-role-name>"
            ]
        }
    ]
}
```

- データカタログでフェデレーティッドカタログを作成するには、データレイク設定 (aws lakeformation get-data-lake-settings) をチェックして、使用している IAM ロールが Lake Formation データレイク管理者であることを確認します。

  データレイク管理者でない場合は、カタログを作成するための Lake Formation CREATE_CATALOG アクセス許可が必要です。次の例は、カタログ作成に必要なアクセス許可を付与する方法を示しています。

```
aws lakeformation grant-permissions \
--cli-input-json \
        '{
            "Principal": {
             "DataLakePrincipalIdentifier":"arn:aws:iam::123456789012:role/non-
admin"
            },
            "Resource": {
                "Catalog": {
                }
            },
            "Permissions": [
                "CREATE_CATALOG",
                "DESCRIBE"
            ]
        }'
```

  アクセスCREATE_CATALOG許可に加えて、登録された接続DATA_LOCATION_ACCESSでフェデレーティッドカタログを作成する必要があります。データレイク管理者には暗黙的なデータロケーション許可があり、この許可は必要ありません。詳細については、「暗黙的な Lake Formation アクセス許可」を参照してください。

  次の例は、登録された接続DATA_LOCATION_ACCESSで を付与する方法を示しています。

```
aws lakeformation grant-permissions \
  --cli-input-json \
    '{
      "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:role/non-admin-
role"
      },
      "Resource": {
```

```
        "DataLocation": {
          "CatalogId": "123456789012",
          "ResourceArn": "arn:aws:glue:us-east-1:123456789012:connection/
connection-name"
        }
      },
      "Permissions": ["DATA_LOCATION_ACCESS"]
    }'
```

2. データソース内のデータを暗号化するためにカスタマーマネージドキーを使用している場合 AWS KMS は、次のキーポリシーをキーに追加します。アカウント番号を有効な AWS アカウント番号に置き換え、ロール名を指定します。データはデフォルトで KMS キーを使用して暗号化されます。Lake Formation には、暗号化用のカスタム KMS キーを作成するオプションがあります。カスタマーマネージドキーを使用している場合は、特定のキーポリシーをキーに追加する必要があります。

   カスタマーマネージドキーの許可を管理する方法の詳細については、カスタマーマネージドキーを参照してください。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:123456789012:key/key-1"
        }
    ]
}
```

AWS Glue 接続を使用したフェデレーティッドカタログの作成

AWS Glue Data Catalog を外部データソースに接続するには、外部データソースとの通信を可能にする接続を使用する必要があります AWS Glue 。コンソール AWS Glue 、Create connection API、Amazon SageMaker Lakehouse コンソールを使用して接続を作成できます AWS Glue 。

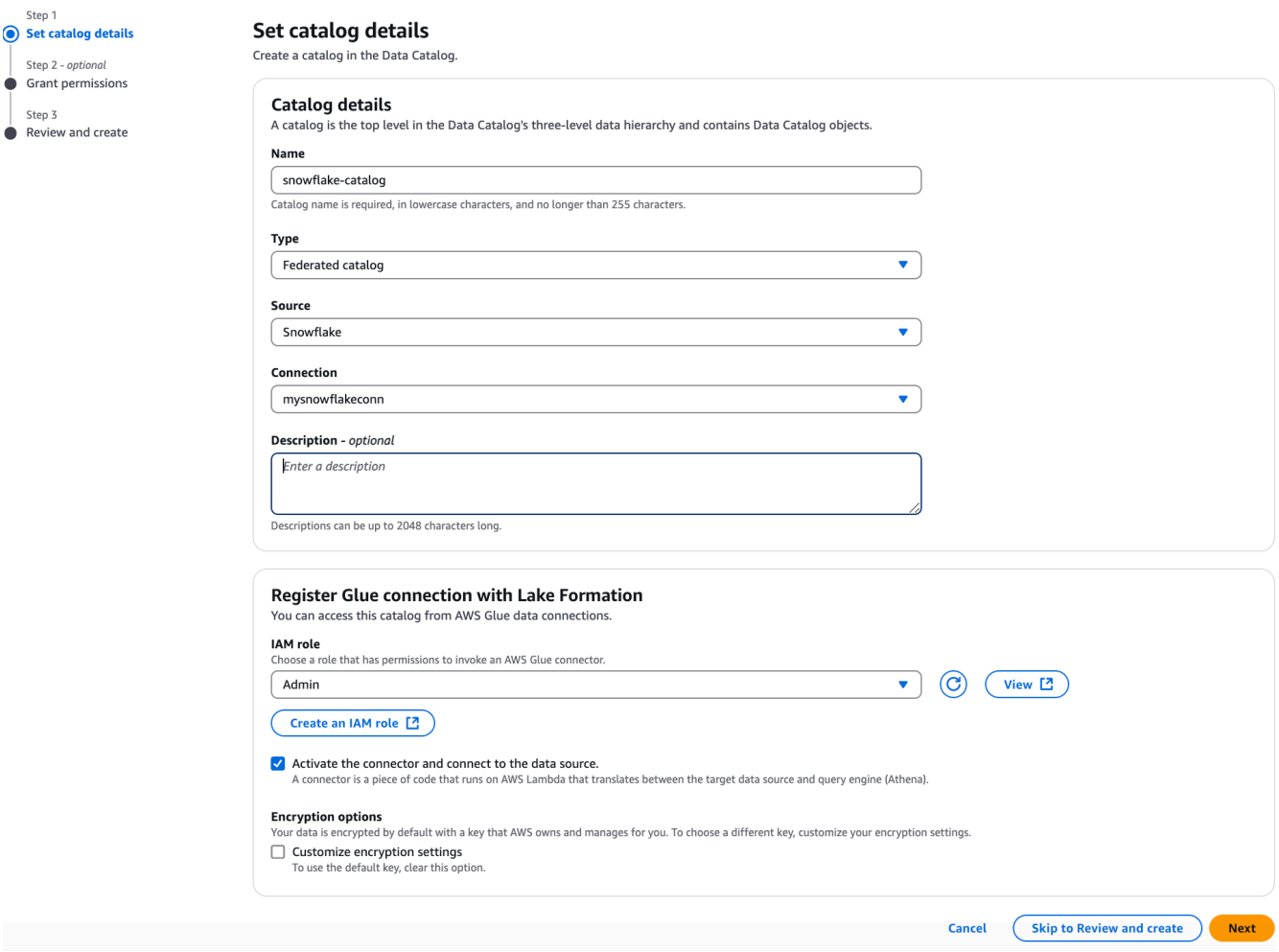
AWS Glue 接続を作成する手順については、「 AWS Glue デベロッパーガイド」の「データへの接続」またはAmazon SageMaker Lakehouse での接続の作成」を参照してください。

ユーザーがフェデレーティッドテーブルに対してクエリを実行すると、Lake Formation は AWS Glue 接続で指定された AWS Lambda 関数を呼び出してデータソースからメタデータオブジェクトを取得する認証情報を提供します。

AWS マネジメントコンソール

外部データソースからフェデレーティッドカタログを作成し、アクセス許可を設定するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。
3. [カタログの作成] オプションを選択します。
4. [カタログの設定] の詳細ページで、次の情報を入力します。



- 名前 – フェデレーティッドカタログの一意の名前。名前は変更できず、小文字にする必要があります。名前は最大 255 文字で、アカウント内で一意である必要があります。
- タイプ – カタログタイプとしてフェデレーティッドカタログを選択します。
- ソース – ドロップダウンからデータソースを選択します。接続を作成したデータソースが表示されます。外部データソース AWS Glue への接続の作成の詳細については、「 デベロッパーガイド」の「コネクタの接続の作成」またはAmazon SageMaker Lakehouse での接続の作成」を参照してください。 AWS Glue
- 接続 — データソースへの既存の AWS Glue 接続を選択します。
- 説明 – データソースから作成されたカタログの説明を入力します。

5. クエリエンジンがデータソースのデータにアクセスするための認証情報を提供するために Lake Formation が引き受ける [IAM ロール] を選択します。このロールには、 AWS Glue 接続にアクセスし、Lambda 関数を呼び出して外部データソースのデータにアクセスするために必要なアクセス許可が必要です。

IAM コンソールで新しいロールを作成することもできます。

必要なアクセス許可については、「データカタログを外部データソースに接続するための前提条件」セクションを参照してください。

6. [コネクタを有効化してデータソースに接続する] オプションを選択して、Athena がフェデレーティッドクエリを実行できるようにします。

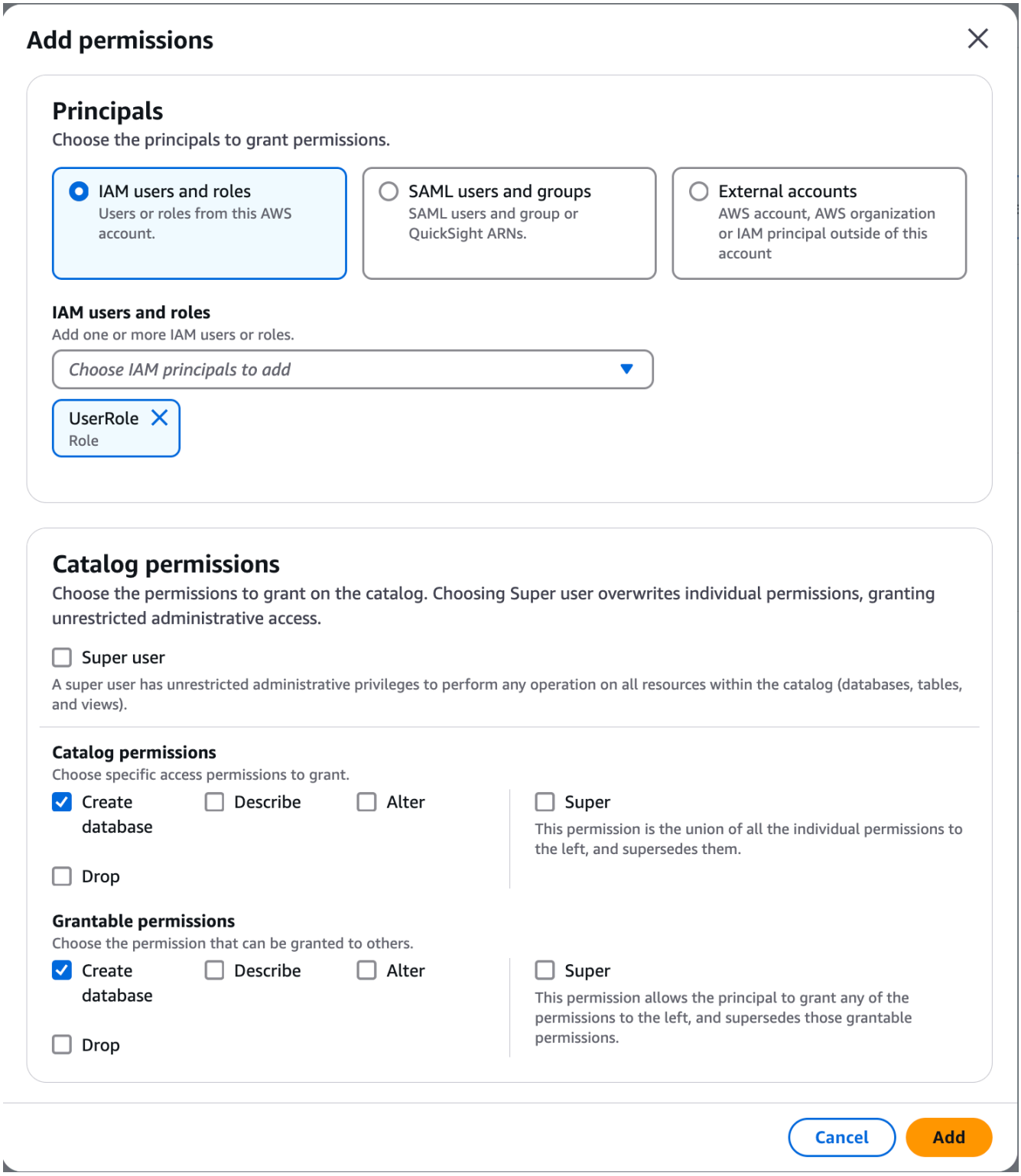
   サポートされているコネクタのリストについては、「Amazon Athena ユーザーガイド」の「接続を登録する」を参照してください。

7. 暗号化オプション – カスタムキーを使用してカタログを暗号化する場合は、[暗号化設定のカスタマイズ] オプションを選択します。カスタムキーを使用するには、KMS キーにカスタムマネージドキーポリシーを追加する必要があります。

8. [次へ] を選択して、他のプリンシパルにアクセス許可を付与します。

9. [アクセス許可を付与] ページで、[アクセス許可を追加] を選択します。

10. [アクセス許可を追加] 画面で、プリンシパルと、付与するアクセス許可のタイプを選択します。



- [Principals] (プリンシパル) セクションでプリンシパルタイプを選択してから、アクセス許可の付与先となるプリンシパルを指定します。

- IAM ユーザーとロール – IAM ユーザーとロールのリストから、ユーザーまたはロールを 1 つ以上選択します。
- SAML ユーザーとグループ – SAML および Amazon Quick ユーザーとグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、または Amazon Quick ユーザーまたはグループの ARNs を入力します。各 ARN の後で [Enter] キーを押します。

- [Permissions] (許可) セクションで、許可と付与可能な許可を選択します。

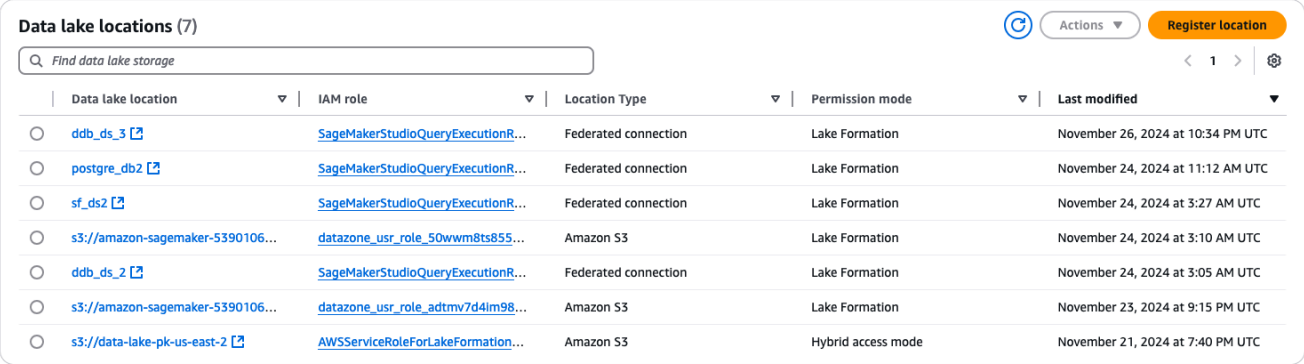
  [カタログのアクセス許可] で、付与するアクセス許可を 1 つ以上選択します。

  カタログ内のすべてのリソースに対する無制限の管理アクセス許可を付与するには、[スーパーユーザー] を選択します。

  付与可能なアクセス許可で、付与受信者が AWS アカウントの他のプリンシパルに付与できるアクセス許可を選択します。このオプションは、外部アカウントから IAM プリンシパルにアクセス許可を付与する場合はサポートされません。

11. [次へ] を選択して情報を確認し、カタログを作成します。[カタログ] リストには、新しいフェデレーティッドカタログが表示されます。

    [データの場所] リストには、新しく登録されたフェデレーティッド接続が表示されます。



AWS CLI

外部データソースからフェデレーティッドカタログを作成し、アクセス許可を設定するには

1. 次の例は、 AWS Glue 接続を作成する方法を示しています。

```
aws glue create-connection
  --connection-input \
```

```
      '{
         "Name": "DynamoDB connection",
         "ConnectionType": "DYNAMODB",
         "Description": "A connection created for DynamoDB",
         "ConnectionProperties": {},
         "AthenaProperties": "spill_prefix": "your_spill_prefix",
         "lambda_function_arn": "Lambda_function_arn",
         "spill_bucket": "Your_Bucket_name",
         "AuthenticationConfiguration": {}
      }'
```

2. 次の例は、Lake Formation に AWS Glue 接続を登録する方法を示しています。

```
aws lakeformation register-resource
  --cli-input-json \
    {"ResourceArn":"arn:aws:glue:us-east-1:123456789012:connection/
dynamo","RoleArn":"arn:aws:iam::123456789012:role/
AdminTelemetry","WithFederation":true}
```

3. データレイク管理者でない場合は、フェデレーティッドカタログを作成するプリンシパルへの登録済み接続DATA_LOCATION_ACCESSに対して を付与します。データレイク管理者には暗黙的なデータロケーション許可があり、この許可は必要ありません。詳細については、「暗黙的な Lake Formation アクセス許可」を参照してください。

   次の例は、登録された接続DATA_LOCATION_ACCESSで を付与する方法を示しています。

```
aws lakeformation grant-permissions \
  --cli-input-json \
    '{
      "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:role/non-
admin-role"
      },
      "Resource": {
        "DataLocation": {
          "CatalogId": "123456789012",
          "ResourceArn": "arn:aws:glue:us-east-1:123456789012:connection/
connection-name"
        }
      },
      "Permissions": ["DATA_LOCATION_ACCESS"]
```

```
    }'
```

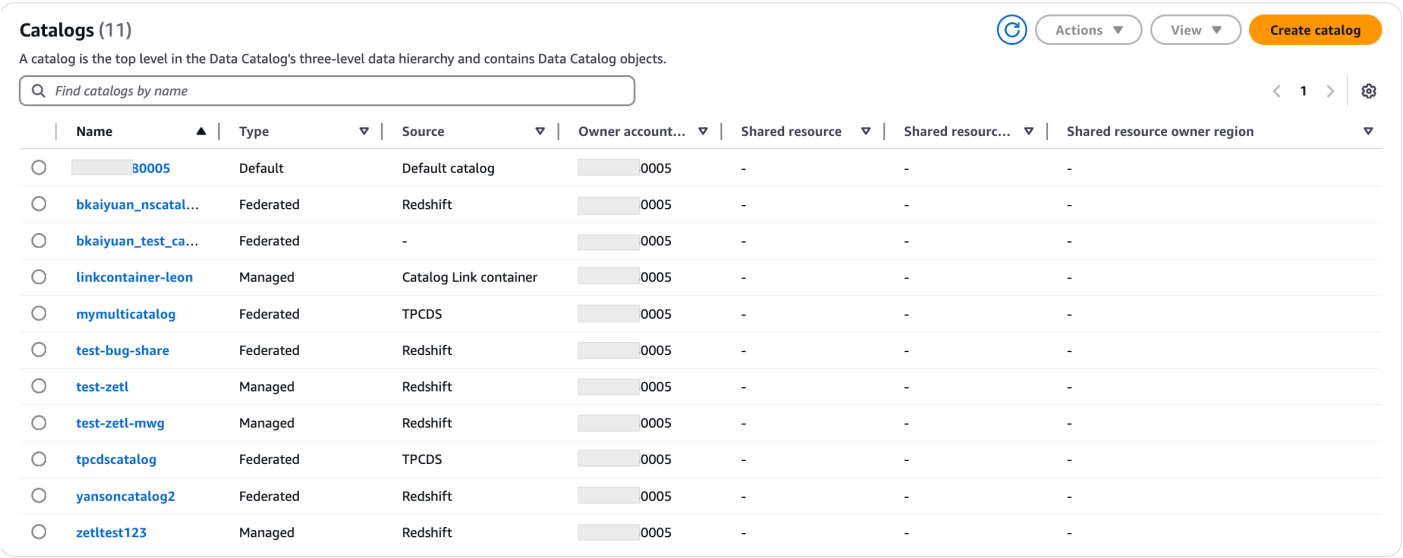
4. 次の例は、フェデレーティッドカタログを作成する方法を示しています。

```
aws glue create-catalog
 --cli-input-json \
      '{
       "Name":"ddbcatalog",
       "CatalogInput":{"CatalogProperties":{"DataLakeAccessProperties":
{"DataTransferRole":"arn:aws:iam::123456789012:role/role name"}},
       "CreateDatabaseDefaultPermissions":[],
       "CreateTableDefaultPermissions":[],
       "FederatedCatalog":{"ConnectionName":"dynamo","Identifier":"dynamo"}
         }
       }'
```

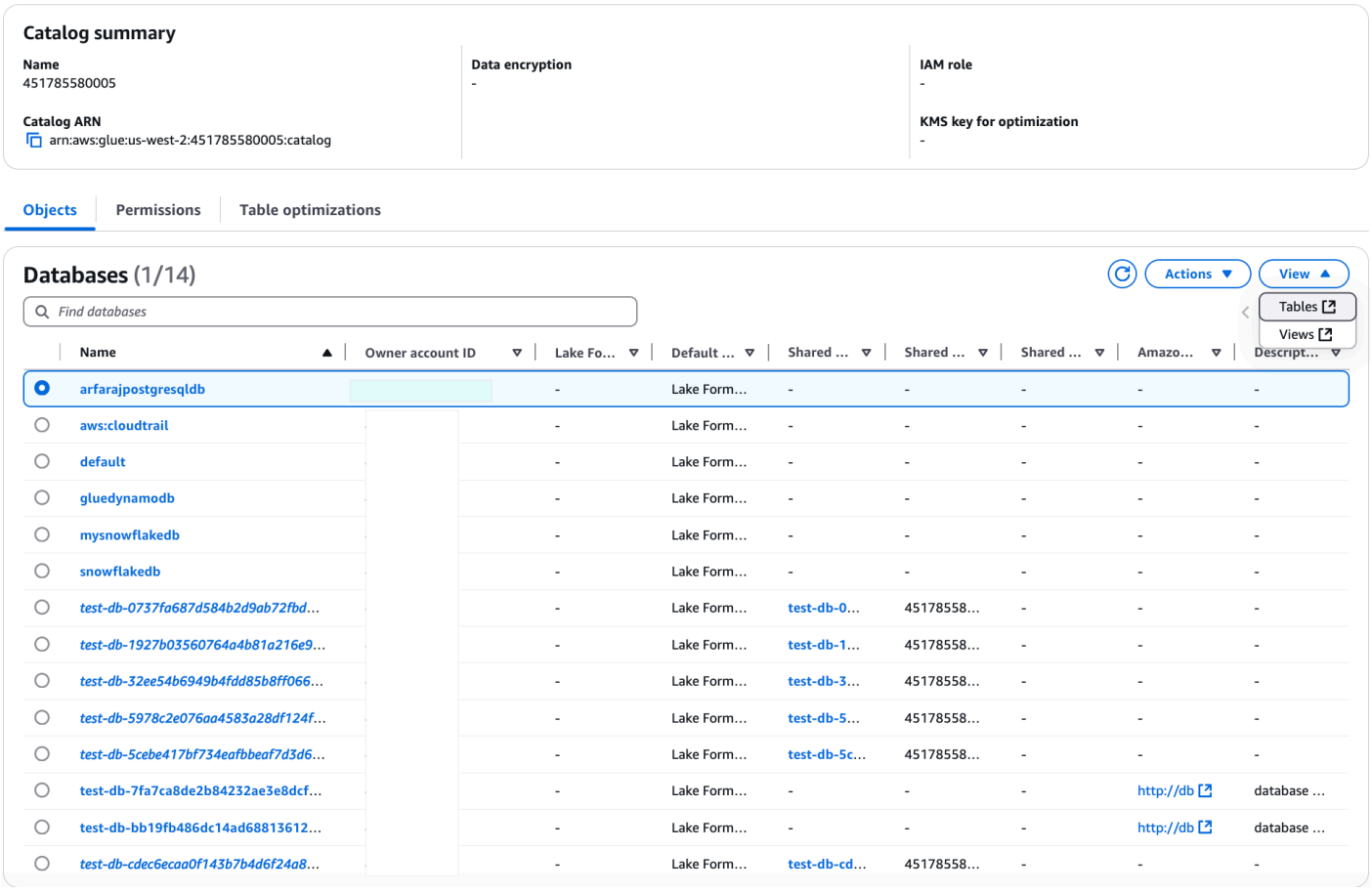
カタログオブジェクトの表示

使用可能なデータソースごとに、 は対応するカタログを に AWS Glue 作成します AWS Glue Data Catalog。カタログを作成したら、Lake Formation コンソールまたは AWS CLIを使用してカタログ内のデータベースとテーブルを表示できます。/ の場合

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. [データカタログ] で [カタログ] を選択します。カタログページには、アクセス許可があるカタログが表示されます。



3. リストからカタログを選択すると、カタログに含まれるデータベースとテーブルが表示されます。リストには、アカウント内のデータベースとリソースリンクも含まれています。これは、外部アカウントの共有データベースとテーブルへのリンクで、データレイク内のデータへのクロスアカウントアクセスに使用されます。



4. [表示] で [テーブル] オプションを選択して、データベース内のテーブルを表示および管理します。

AWS CLI カタログとデータベースを表示するための例

次の例は、 を使用してカタログを表示する方法を示しています。 AWS CLI

```
aws glue get-catalog \
--catalog-id 123456789012:dynamodbcatalog
```

次の例は、アカウント内のすべてのカタログをリクエストする方法を示しています。

```
aws glue get-catalogs \
```

```
 --recursive
```

次のリクエスト例は、カタログでデータベースを取得する方法を示しています。

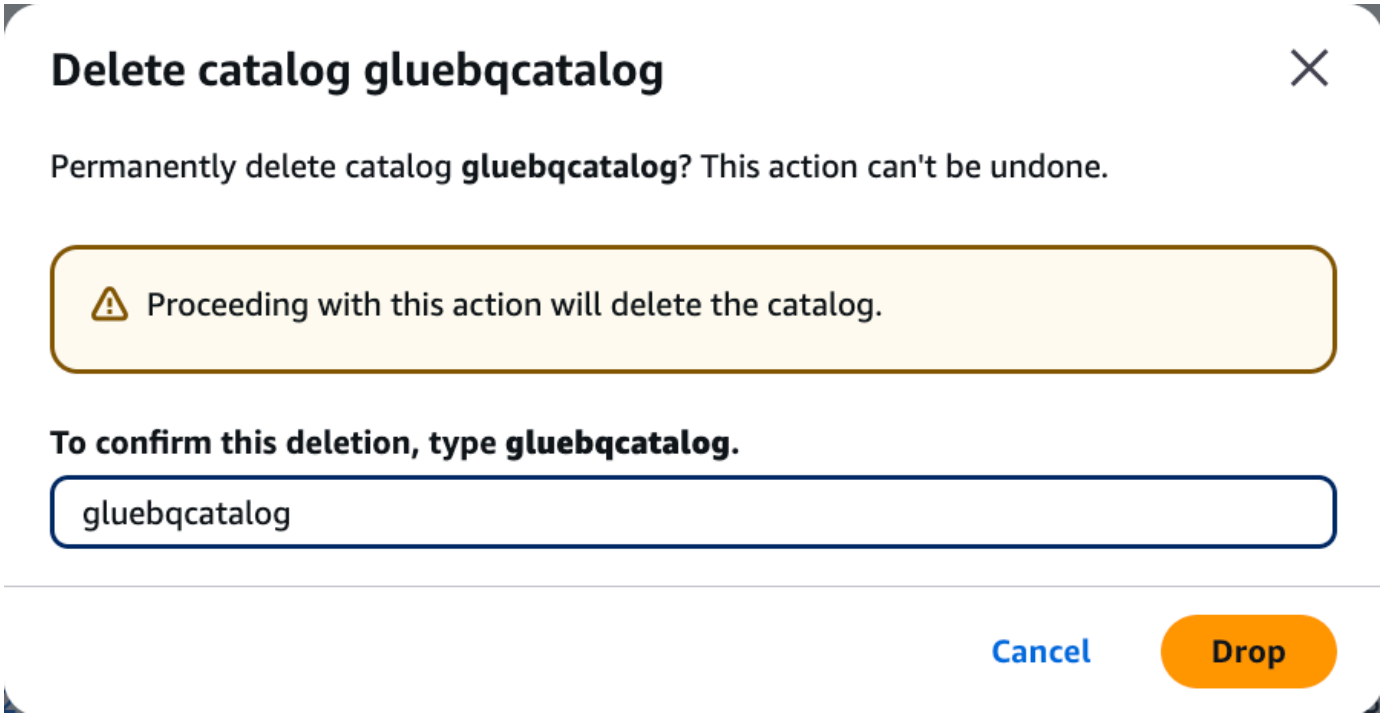
```
aws glue get-database \
--catalog-id 123456789012:dynamodbcatalog
--database-name database name
```

フェデレーティッドカタログの削除

glue:DeleteCatalog オペレーションまたは AWS Lake Formation コンソール AWS Glue Data Catalog を使用して、 で作成したフェデレーティッドカタログを削除できます。

フェデレーティッドカタログを削除するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。
3. カタログリストから削除するカタログを選択します。
4. [アクション] で、[削除] を選択します。
5. [ドロップ] を選択して確認すると、データカタログからフェデレーティッドカタログが削除されています。



フェデレーティッドカタログを削除するには (CLI)

- ```
  aws glue delete-catalog
    --catalog-id 123456789012:catalog name
  ```

フェデレーティッドカタログのクエリ

他のプリンシパルにアクセス許可を付与すると、プリンシパルはサインインし、Athena を使用してフェデレーティッドカタログ内のテーブルのクエリを開始できます。

フェデレーティッドデータベースでテーブルを作成および削除するには、プリンシパルに Lake Formation の `Create table` と `Drop` のアクセス許可が必要です。

データカタログアクセス許可を付与する方法の詳細については、「データカタログリソースに対するアクセス許可の付与」を参照してください。

データカタログのクエリの詳細については Amazon Athena、Amazon Athena ユーザーガイド」の「AWS Glue Data Catalog からのクエリ Amazon Athena」を参照してください。

その他のリソース

このブログ記事では、データアナリストが Amazon Redshift データウェアハウスや Amazon DynamoDB データベースなど、S3 データレイクの外部に保存されているデータに、単一の統一されたエクスペリエンスで安全にアクセスしてクエリを実行する方法について説明します。管理者は、さまざまな詳細レベルでアクセスコントロールを適用して、データアクセスを拡張しながら機密データを確実に保護できるようになりました。これにより、組織はセキュリティとコンプライアンスを維持しながらデータイニシアチブを加速できるため、より迅速なデータ主導の意思決定が可能になります。

- Amazon SageMaker Lakehouse を使用した Amazon Athena フェデレーティッドクエリのカタログ化と管理

Amazon S3 Tables と AWS Glue Data Catalog および の統合 AWS Lake Formation

Amazon S3 Tables は、分析ワークロードに特別に最適化された S3 ストレージを提供し、クエリのパフォーマンスを向上させながらコストを削減します。S3 Tables のデータは、新しいバケットタイ
  ```

プ、つまりテーブルをサブリソースとして保存するテーブルバケットに保存されます。S3 テーブルには Apache Iceberg 標準のサポートが組み込まれているため、Apache Spark などの一般的なクエリエンジンを使用して、Amazon S3 テーブルバケット内の表形式データに対して簡単にクエリを実行できます。

Amazon S3 Tables は、IAM アクセスコントロールまたは IAM および Lake Formation 許可 AWS Glue Data Catalog を使用して と統合できます。

- IAM アクセスコントロール: IAM ポリシーを使用して S3 テーブルとデータカタログへのアクセスを制御します。このアクセスコントロールアプローチでは、リソースにアクセスするには、S3 Tables リソースと Data Catalog オブジェクトの両方に対する IAM アクセス許可が必要です。
- Lake Formation アクセスコントロール: データカタログを介した S3 テーブルへのアクセスを制御するために AWS Glue 、IAM アクセス許可に加えて AWS Lake Formation 許可を使用します。このモードでは、プリンシパルは Data Catalog を操作するための IAM アクセス許可を必要とし、Lake Formation 許可によってプリンシパルがアクセスできるカタログリソース (データベース、テーブル、列、行) が決まります。このモードは、粗粒度のアクセスコントロール (データベースレベルとテーブルレベルの権限) ときめ細かなアクセスコントロール (列レベルと行レベルのセキュリティ) の両方をサポートします。登録されたロールが設定され、認証情報供給が有効になっている場合、Lake Formation は登録されたロールを使用してプリンシパルに代わって認証情報を提供するため、プリンシパルに S3 Tables IAM アクセス許可は必要ありません。Lake Formation アクセスコントロールは、サードパーティーの分析エンジンの認証情報供給もサポートしています。

このセクションでは、以下のシナリオ AWS Lake Formation で との統合を設定するためのガイダンスを提供します。

- シナリオ A: IAM アクセスコントロールを使用して S3 テーブルとデータカタログを統合したため、使用する予定です AWS Lake Formation。詳細については、「S3 Tables 統合のアクセスコントロールの変更」を参照してください。
- シナリオ B: を使用して S3 テーブルとデータカタログを統合する予定 AWS Lake Formation で、現在アカウントとリージョンに統合されていません。Amazon S3 テーブルカタログを データカタログおよび Lake Formation と統合するための前提条件 セクションから開始し、 に従いますAmazon S3 Tables 統合の有効化。
- シナリオ C: を使用して S3 テーブルとデータカタログを統合し AWS Lake Formation 、IAM を使用する予定です。詳細については、「S3 Tables 統合のアクセスコントロールの変更」を参照してください。

S3 Tables と AWS 分析サービスとの統合の手順に従って、 AWS Glue Data Catalog とテーブルリソースにアクセスし、 AWS 分析サービスと連携するための適切なアクセス許可があることを確認してください。

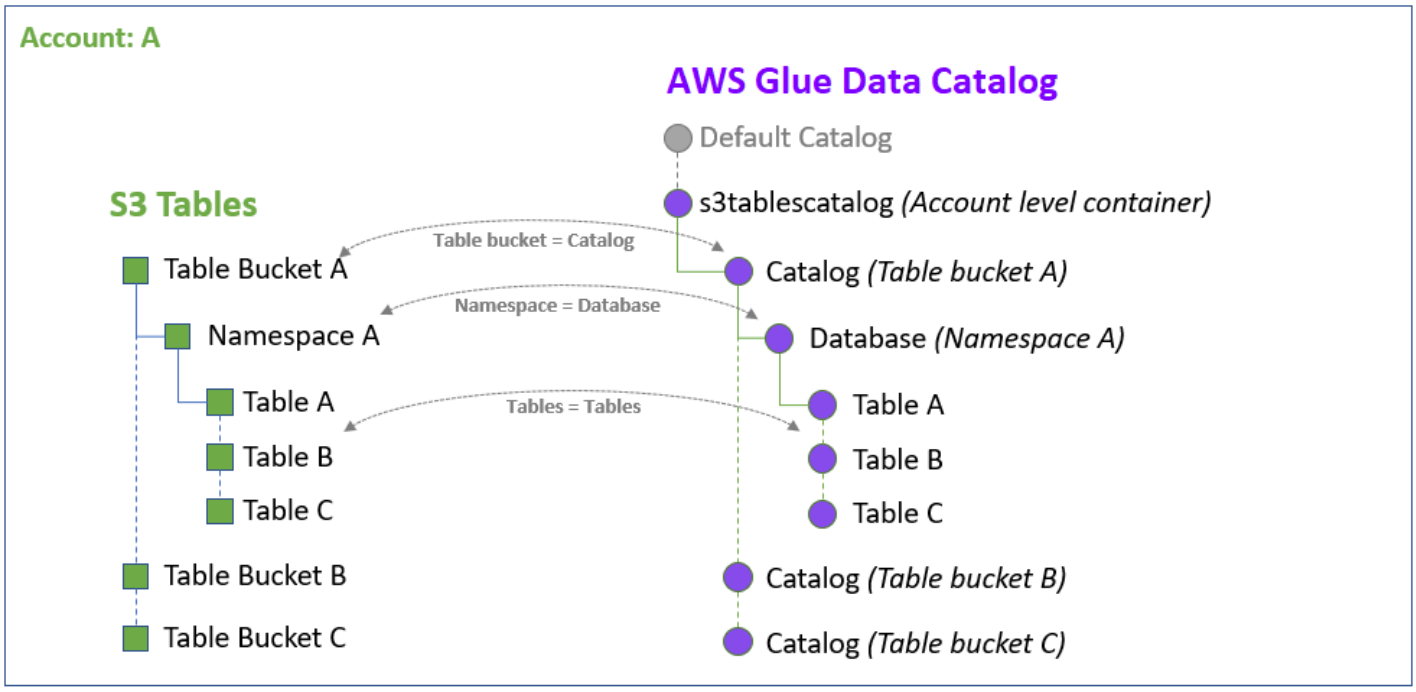
トピック

- データカタログと Lake Formation の統合の仕組み
- Amazon S3 テーブルカタログを データカタログおよび Lake Formation と統合するための前提条件
- Amazon S3 Tables 統合の有効化
- S3 テーブルカタログでのデータベースとテーブルの作成
- 別の AWS アカウントに Amazon S3 テーブルバケットを登録する
- アクセス許可の付与

データカタログと Lake Formation の統合の仕組み

S3 テーブルカタログを データカタログおよび Lake Formation と統合すると、 AWS Glue サービスは AWS リージョンに固有のアカウントのデフォルトデータカタログに `s3tablescatalog` という名前の単一のフェデレーティッドカタログを作成します。統合では、アカウントとフェデレーティッドカタログ内のすべての Amazon S3 テーブルバケットリソース AWS リージョン を次の方法でマッピングします。

- Amazon S3 テーブルバケットは、データカタログ内のマルチレベルカタログになります。
- 関連付けられた Amazon S3 名前空間は、データカタログにデータベースとして登録されます。
- テーブルバケット内の Amazon S3 Tables は、データカタログ内のテーブルになります。



Lake Formation と統合した後、テーブルバケットカタログに Apache Iceberg テーブルを作成し、Amazon Athena Amazon EMR やサードパーティー AWS の分析エンジンなどの統合分析エンジンを介してそれらにアクセスできます。

統合で Lake Formation も有効にすると、きめ細かなアクセスコントロールが可能になります AWS Lake Formation。このセキュリティアプローチは、 (IAM) アクセス許可に加えて AWS Identity and Access Management 、テーブルを操作する前に、テーブルに対する Lake Formation アクセス許可を IAM プリンシパルに付与する必要があることを意味します。

AWS Lake Formationには、2 つの主な許可タイプがあります。

- メタデータに対するアクセス許可は、データカタログ内でメタデータデータベースとテーブルを作成、読み取り、更新、削除できるかどうかを制御します。
- 基になるデータに対するアクセス許可は、データカタログリソースの参照先となる Amazon S3 の場所にデータを読み書きできるかどうかを制御します。

Lake Formation では、独自のアクセス許可モデルと IAM アクセス許可モデルを組み合わせて使用して、データカタログリソースと基になるデータへのアクセスを制御します。

- データカタログリソースまたは基になるデータへのアクセスリクエストが成功するには、そのリクエストが IAM と Lake Formation の両方によるアクセス許可のチェックに合格する必要があります。
- IAM アクセス許可は Lake Formation および AWS Glue APIs とリソースへのアクセスを制御し、Lake Formation アクセス許可は Data Catalog リソース、Amazon S3 ロケーション、および基盤となるデータへのアクセスを制御します。

Lake Formation 許可は付与されたリージョン内でのみ適用されます。プリンシパルが Lake Formation 許可を付与されるには、データレイク管理者または必要なアクセス許可を持つ別のプリンシパルによって認可される必要があります。

Amazon S3 テーブルカタログを データカタログおよび Lake Formation と統合するための前提条件

以下は、Amazon S3 テーブルと AWS Glue Data Catalog および の統合を有効にするための前提条件です AWS Lake Formation。

1. Amazon S3 Tables の統合を有効にすると、Lake Formation は S3 テーブルの場所を自動的に登録します。Lake Formation にテーブルバケットの場所を登録するには、`lakeformation:RegisterResource`、`lakeformation:RegisterResourceWithPrivileg` および `lakeformation:CreateCatalog` のアクセス許可を持つ IAM ロール/ユーザーが必要です。管理者以外のこれらのアクセス許可を持つユーザーがカタログの場所を登録すると、Lake Formation によってその場所に対する `DATA_LOCATION_ACCESS` アクセス許可が自動的に付与され、呼び出し元のプリンシパルに、登録されたデータの場所でサポートされているすべての Lake Formation オペレーションを実行するアクセス許可が付与されます。
2. S3 テーブル統合を有効にする場合は、データアクセスを許可する認証情報を提供する Lake Formation の IAM ロールを選択する必要があります。S3 テーブルバケットへの Lake Formation データアクセス用の IAM ロールを作成します。テーブルバケットを Lake Formation に登録する際に使用する IAM ロールには、次のアクセス許可が必要です。

   JSON

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

```
            "Sid": "LakeFormationPermissionsForS3ListTableBucket",
            "Effect": "Allow",
            "Action": [
                "s3tables:ListTableBuckets"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "LakeFormationDataAccessPermissionsForS3TableBucket",
            "Effect": "Allow",
            "Action": [
                "s3tables:CreateTableBucket",
                "s3tables:GetTableBucket",
                "s3tables:CreateNamespace",
                "s3tables:GetNamespace",
                "s3tables:ListNamespaces",
                "s3tables:DeleteNamespace",
                "s3tables:DeleteTableBucket",
                "s3tables:CreateTable",
                "s3tables:DeleteTable",
                "s3tables:GetTable",
                "s3tables:ListTables",
                "s3tables:RenameTable",
                "s3tables:UpdateTableMetadataLocation",
                "s3tables:GetTableMetadataLocation",
                "s3tables:GetTableData",
                "s3tables:PutTableData"
            ],
            "Resource": [
                "arn:aws:s3tables:us-east-1:123456789012:bucket/*"
            ]
        }
    ]
}
```

詳細については、「[ロケーションの登録に使用されるロールの要件]」を参照してください。

3. Lake Formation サービスでロールを引き受け、統合された分析エンジンに一時的な認証情報を提供できるようにするには、IAM ロールに次の信頼ポリシーを追加します。

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "Service": "lakeformation.amazonaws.com"
  },
  "Action": [
    "sts:AssumeRole",
    "sts:SetSourceIdentity",
    "sts:SetContext"  # add action to trust relationship when using IAM Identity
 center principals with Lake Formation
  ]
}
```

> **Note**
>
> AWS 分析サービスの統合プロセスが更新されました。プレビューリリースで統合を設定した場合は、現在の統合を引き続き使用できます。ただし、更新された統合プロセスではパフォーマンスが向上します。統合を更新するには:
>
> 1. まず、Lake Formation で既存の S3 テーブルカタログを削除します。カタログを削除するには、カタログリストから `s3tablescatalog` カタログを選択し、[アクション] から [削除] を選択します。
> 2. 次に、`s3tablescatalog` のデータの場所を登録解除します。
>   a. Lake Formation コンソールの [管理] セクションで、[データの場所] を選択します。
>   b. 場所を選択し、[アクション] メニューで [削除] を選択します。
>   c. 確認を求めるプロンプトが表示されたら、[Remove] (削除) を選択します。
>
>   データロケーションの登録解除の詳細な手順については、Amazon S3 ロケーションの登録解除「」セクションを参照してください。
> 3. 次に、Amazon S3 Tables 統合の有効化「」セクションの更新された統合ステップに従います。

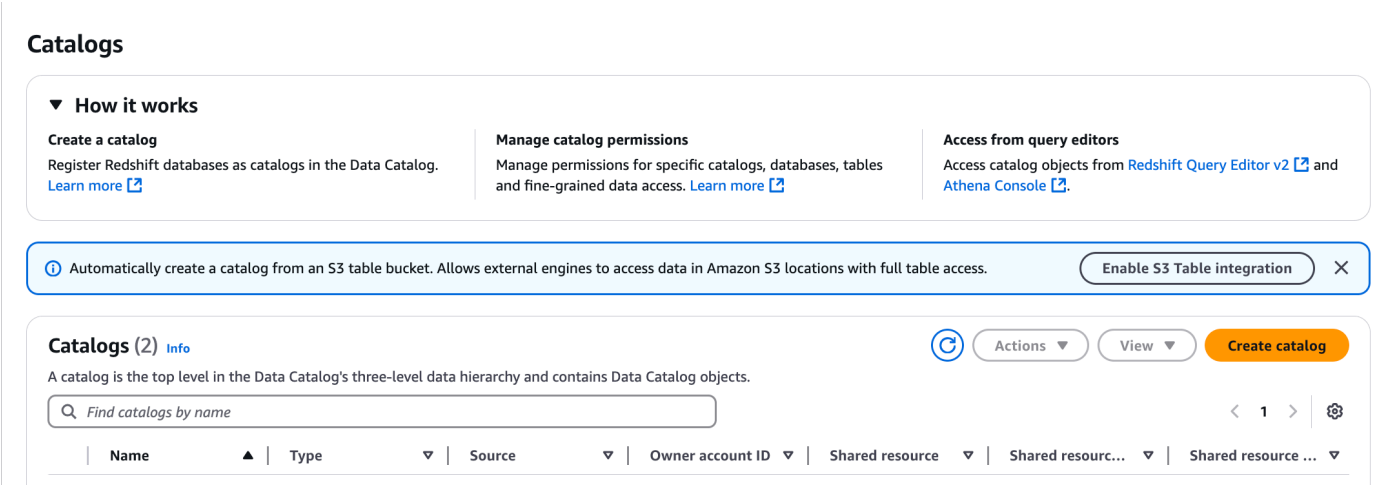
Amazon S3 Tables 統合の有効化

Amazon S3 コンソールを使用して Amazon S3 テーブルバケットを作成し、 AWS 分析サービスと統合できます。詳細については、「Amazon S3 Tables と AWS 分析サービスの統合」を参照してください。

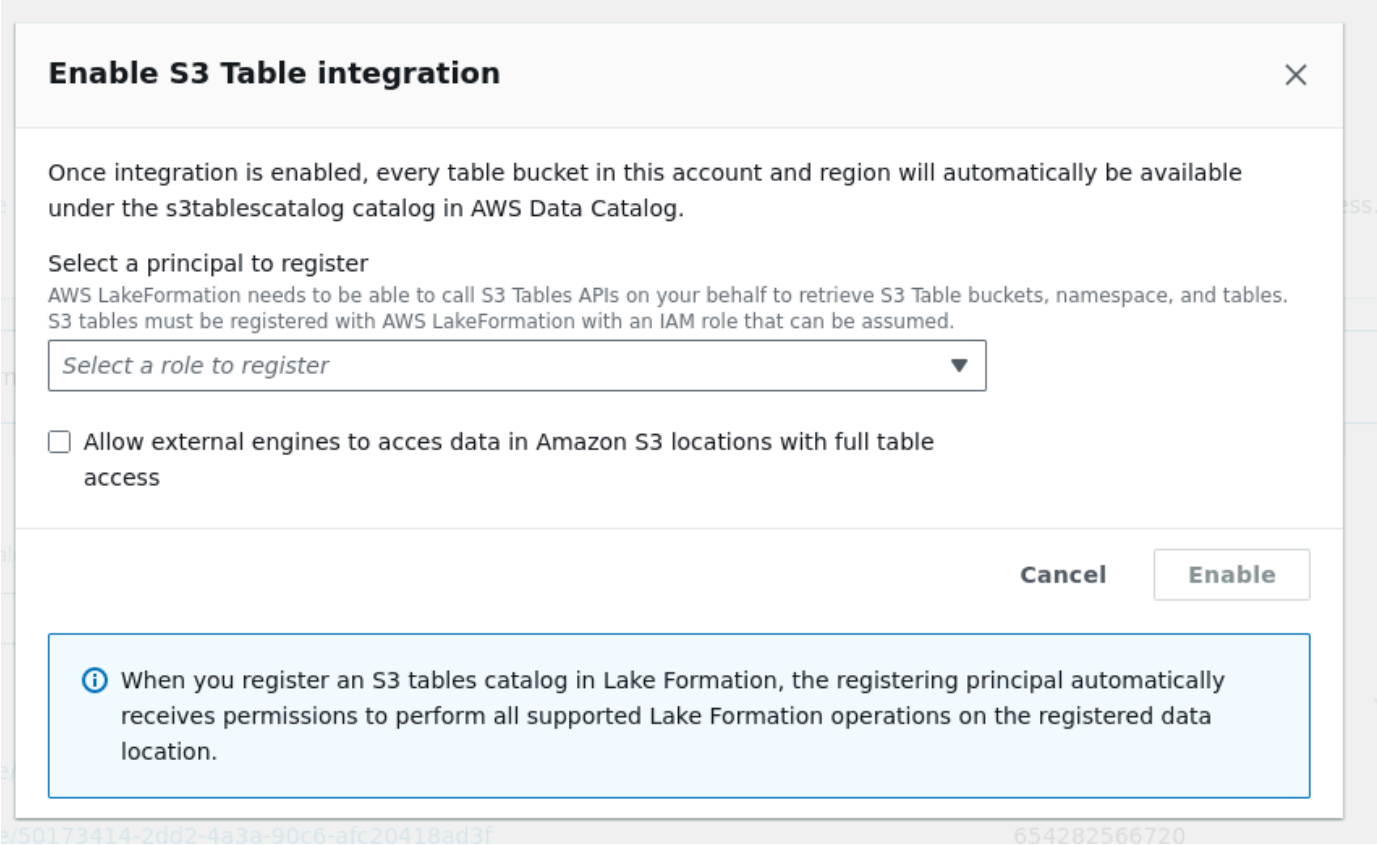
では AWS Lake Formation、Lake Formation コンソールまたは AWS Lake Formation を使用して、Amazon S3 Tables と AWS Glue Data Catalog および との統合を有効にできます AWS CLI。

Amazon S3 Tables をデータカタログおよび Lake Formation と統合するには (コンソール)

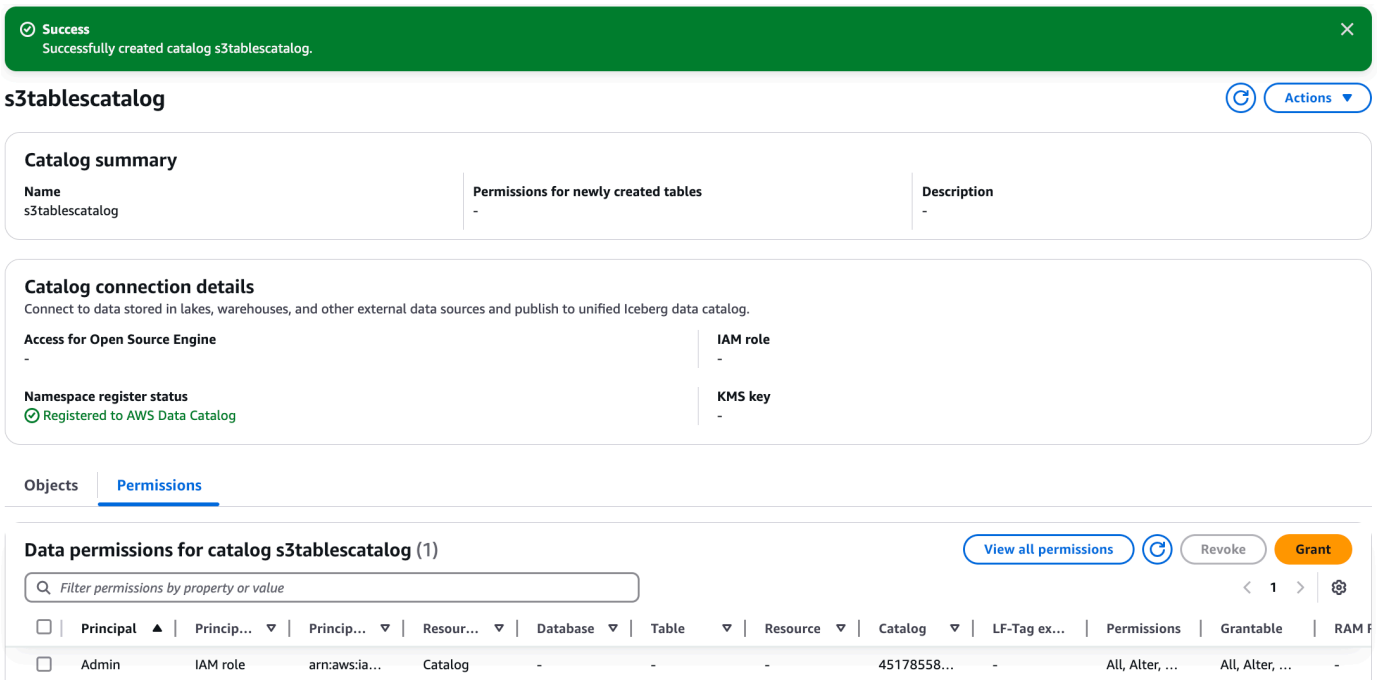
1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。
3. [カタログ] ページで [S3 テーブル統合を有効にする] を選択します。



4. Lake Formation が分析クエリエンジンに認証情報を提供するために必要なアクセス許可を持つ IAM ロールを選択します。ロールがデータにアクセスするために必要なアクセス許可については、「前提条件」セクションの「step3-permissions」を参照してください。



5. [外部エンジンに、Amazon S3 の場所のデータに対するフルアクセス許可を付与する] を選択します。サードパーティーエンジンのフルテーブルアクセスを有効にすると、Lake Formation は IAM セッションタグの検証を実行せずにサードパーティーエンジンに直接認証情報を返します。つまり、アクセスするテーブルに Lake Formation のきめ細かなアクセスコントロールを適用することはできません。
6. [有効化] を選択します。S3 Tables の新しいカタログがカタログリストに追加されます。S3 テーブルカタログ統合を有効にすると、サービスは S3 テーブルバケットのデータロの場所を Lake Formation に登録します。
7. カタログを選択してカタログオブジェクトを表示し、他のプリンシパルにアクセス許可を付与します。



マルチレベルカタログを作成するには、「Amazon Simple Storage Service ユーザーガイド」の「テーブルバケットの作成」セクションを参照してください。

Amazon S3 Tables をデータカタログおよび Lake Formationと統合するには (CLI)

前提条件セクションに従って、Lake Formation がテーブルリソースにアクセスできるようにする IAM サービスロールを作成します。

1. 次の信頼ポリシーを含むファイルを `Role-Trust-Policy.json` という名前で作成します。

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

```
        "Condition": {
          "StringEquals": {
            "aws:SourceAccount": "111122223333"
          }
        }
      }
    ]
}
```

2. 次のコマンドを使用して IAM サービスロールを作成します。

```
aws iam create-role \
  --role-name S3TablesRoleForLakeFormation \
  --assume-role-policy-document file://Role-Trust-Policy.json
```

3. 次のポリシーを含むファイルを LF-GluePolicy.json という名前で作成します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LakeFormationPermissionsForS3ListTableBucket",
            "Effect": "Allow",
            "Action": [
                "s3tables:ListTableBuckets"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "LakeFormationDataAccessPermissionsForS3TableBucket",
            "Effect": "Allow",
            "Action": [
                "s3tables:CreateTableBucket",
                "s3tables:GetTableBucket",
                "s3tables:CreateNamespace",
                "s3tables:GetNamespace",
                "s3tables:ListNamespaces",
                "s3tables:DeleteNamespace",
                "s3tables:DeleteTableBucket",
                "s3tables:CreateTable",
                "s3tables:DeleteTable",
```

```
                "s3tables:GetTable",
                "s3tables:ListTables",
                "s3tables:RenameTable",
                "s3tables:UpdateTableMetadataLocation",
                "s3tables:GetTableMetadataLocation",
                "s3tables:GetTableData",
                "s3tables:PutTableData"
            ],
            "Resource": [
                "arn:aws:s3tables:us-east-1:111122223333:bucket/*"
            ]
        }
    ]
}
```

4. 次のコマンドを使用してポリシーをロールにアタッチします。

```
aws iam put-role-policy \
  --role-name S3TablesRoleForLakeFormation \
  --policy-name LakeFormationDataAccessPermissionsForS3TableBucket \
  --policy-document file://LF-GluePolicy.json
```

5. 以下を含むファイルを input.json という名前で作成します。

```
{
    "ResourceArn": "arn:aws:s3tables:us-east-1:111122223333:bucket/*",
    "WithFederation": true,
    "RoleArn": "arn:aws:iam::111122223333:role/S3TablesRoleForLakeFormation"
}
```

6. 次のコマンドを使用してテーブルバケットを Lake Formation に登録します。

```
aws lakeformation register-resource \
  --region us-east-1 \
  --with-privileged-access \
  --cli-input-json file://input.json
```

7. 次のカタログを含むファイルを catalog.json という名前で作成します。

```
{
   "Name": "s3tablescatalog",
   "CatalogInput": {
      "FederatedCatalog": {
```

```
            "Identifier": "arn:aws:s3tables:us-east-1:111122223333:bucket/*",
            "ConnectionName": "aws:s3tables"
        },
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": [],
        "AllowFullTableExternalDataAccess": "True"
    }
}
```

8. 次のコマンドを使用して s3tablescatalog カタログを作成します。このカタログを作成すると、テーブルバケット、名前空間、テーブルに対応するオブジェクト AWS Glue Data Catalog が に入力されます。

```
aws glue create-catalog \
  --region us-east-1 \
  --cli-input-json file://catalog.json
```

9. 次のコマンド AWS Glue を使用して、s3tablescatalogカタログが に追加されていることを確認します。

```
aws glue get-catalog --catalog-id s3tablescatalog
```

S3 テーブルカタログでのデータベースとテーブルの作成

データベースを作成して Apache Iceberg テーブルを整理し、テーブルを作成して S3 テーブルカタログ内のデータのスキーマと場所を定義できます。

データベースの作成 (コンソール)

1. コンソール (https://console.aws.amazon.com/lakeformation/) を開き、データレイク管理者またはデータベース作成者としてサインインします。
2. ナビゲーションペインの [データカタログ] で [データベース] を選択します。
3. [データベースの作成] を選択します。
4. [データベースの作成] ページで、[データベース] オプションを選択し、次の詳細を入力します。

   - 名前 – データベース用に一意の名前を入力します。
   - データカタログ – S3 テーブルカタログを選択します。データベースはこのカタログにあります。
   - 説明 – (オプション) 説明と場所を追加します。

- 新しいテーブルの IAM アクセスコントロール – オプションで、[このデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する] を選択します。このオプションの詳細については、「データレイクのデフォルトセキュリティ設定の変更」を参照してください。
- [データベースを作成] を選択します。S3 テーブルカタログの下に作成されたデータベースが表示されます。

を使用してデータベースを作成する AWS CLI

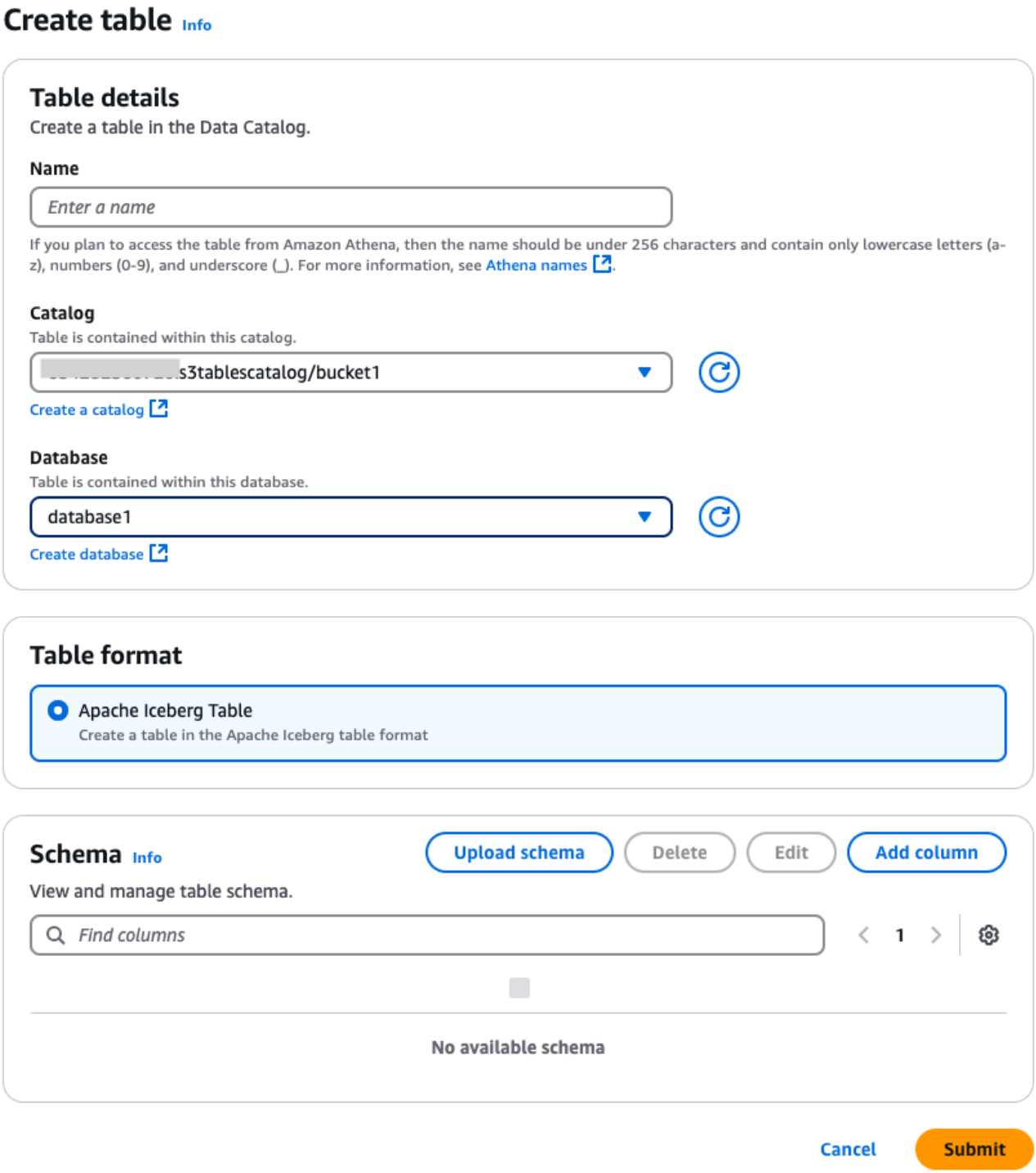
次の CLI コマンドは、S3 テーブルカタログでデータベースを作成する方法を示しています。

```
aws glue create-database
--region us-east-1 \
--catalog-id "123456789012:s3tablescatalog/test" \
--database-input \
 '{ "Name": "testglueclidbcreation" }'
```

テーブルを作成する (AWS マネジメントコンソール)

Lake Formation コンソールまたは AWS Glue `CreateTable` API を使用して、S3 テーブルカタログで Apache Iceberg メタデータテーブルを作成できます。

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開き、データレイク管理者または `CreateTable` アクセス許可を持つユーザーとしてサインインします。
2. ナビゲーションペインの [データカタログ] で、[テーブル] を選択します。
3. [テーブルの作成] を選択します。
4. [テーブルの作成] ページで、テーブルの詳細を入力します。



- テーブル名 – テーブルの一意の名前を入力します。
- カタログ – カタログとして S3 テーブルカタログを選択します。
- データベース – S3 テーブルカタログでデータベースを選択します。

- 説明 – テーブルの説明を入力します。
- スキーマ - [列を追加] を選択して、列と、列のデータ型を追加します。空のテーブルを作成して、後でスキーマを更新することもできます。Iceberg では、テーブルを作成した後でスキーマとパーティションを進化させることができます。[Athena クエリ] を使用してテーブルスキーマを更新し、[Spark クエリ] を使用してパーティションを更新できます。

5. [Submit] を選択してください。

テーブルを作成する (AWS CLI)

```
aws glue create-table \
--database-name "testglueclidbcreation" \
--catalog-id "123456789012:s3tablescatalog/test" \
--region us-east-1 \
--table-input \
'{ "Name": "testtablegluecli", "Parameters": { "format": "ICEBERG" },
 "StorageDescriptor": { "Columns": [ {"Name": "x", "Type": "int", "Parameters":
 {"required": "true"}} ] } }'
```

別の AWS アカウントに Amazon S3 テーブルバケットを登録する

ある AWS アカウントから個々の Amazon S3 テーブルバケット (ARN 形式:`arn:aws:s3tables:us-east-1:`*`account-id`*`:bucket/`*`bucket-name`*) を別のアカウントの Lake Formation に登録できます。例えば、アカウント A のテーブルバケットをアカウント B の Lake Formation に登録できます。

前提条件

クロスアカウントバケット登録を開始する前に:

- アカウント A でテーブルバケットを作成します。
- バケット登録の適切なアクセス許可を持つ IAM ロールをアカウント B に作成します。

  Lake Formation にテーブルバケットを登録するために必要なアクセス許可の詳細については、「Amazon S3 テーブルカタログを データカタログおよび Lake Formation と統合するための前提条件」を参照してください。
- S3 テーブルカタログを作成するアカウントにテーブルバケットを登録します。

- クロスアカウントアクセスの場合、テーブルバケットの登録時に指定されたロールは、クロスアカウントバケットにアクセスするための適切なアクセス許可を持つアカウント内ロールである必要があります。ロールには、バケットリソースにアクセスするために必要な S3 Tables IAM アクションが必要です。

クロスアカウントテーブルバケット登録

次の手順では、アカウント A がリソース所有アカウントであり、アカウント B がアクセス許可を管理するためにテーブルバケットが登録される場所です。

1. アカウント A AWS マネジメントコンソール の にサインインします。

   Amazon S3 コンソール (https://console.aws.amazon.com/s3/) を開きます。

2. テーブルバケットを作成します。詳細については、「Amazon S3 ユーザーガイド」の「テーブルバケットの作成」を参照してください。

3. アカウント B にテーブルバケットを登録します。

   AWS CLI を使用して、アカウント A のテーブルバケットをアカウント B の Lake Formation に登録します。

   ```
   aws lakeformation register-resource \
   --resource-arn 'arn:aws:s3tables:us-east-1:account-A-id:bucket/single-bucket-name'
    \
   --role-arn arn:aws:iam::account-B-id:role/role-name \
   --region us-east-1
   ```

   account-A-id、single-bucket-name、account-B-id、および role-name を特定の値に置き換えます。

4. 次に、アカウント B でテーブルバケットのカタログを作成します。

   AWS CLIを使用してカタログを作成します。

   ```
   aws glue create-catalog --region us-east-1 \
   --cli-input-json \
   '{
      "Name": "catalog-name",
      "CatalogInput" : {
         "FederatedCatalog": {
   ```

```
          "Identifier": "arn:aws:s3tables:us-east-1:account A:bucket/single-bucket-
name",
          "ConnectionName": "aws:s3tables"
       },
       "CreateDatabaseDefaultPermissions": [],
       "CreateTableDefaultPermissions": []
    }
}'
```

catalog-name、account-A-id、bucket-name を特定の値に置き換えます。

次の CLI の例は、カタログの詳細を表示する方法を示しています。

```
aws glue get-catalog \
  --catalog-id account-id:catalog-name \
  --region us-east-1
```

5. 次に、アカウント B で新しく作成されたカタログにデータベースとテーブルを作成します。

データベースを作成します。

```
aws glue create-database \
  --region us-east-1 \
  --catalog-id "account-B-id:catalog-name" \
  --database-input \
'{
  "Name": "database-name"
}'
```

テーブルを作成します。

```
aws glue create-table \
  --database-name database-name \
  --catalog-id account-B-id:catalog-name\
  --region us-east-1 \
  --table-input \
  '{
        "Name": "table-name",
        "Parameters": {
            "format": "ICEBERG"
        },
```

```
        "StorageDescriptor": {
           "Columns": [
        {"Name": "x", "Type": "int", "Parameters": {"required": "true"}}
          ]
        }
}'
```

database-name、account-B-id、catalog-name、table-name を特定の値に置き換えます。

6. 次の例は、カタログ内のオブジェクトを表示する方法を示しています。

   データベースの詳細を表示します。

```
aws glue get-database \
  --name database-name \
  --catalog-id account-B-id:catalog-name \
  --region us-east-1
```

   テーブルの詳細を表示します。

```
aws glue get-table \
  --name table-name \
  --database-name database-name \
  --catalog-id account-B-id:catalog-name \
  --region us-east-1
```

   database-name、account-B-id、catalog-name、table-name を特定の値に置き換えます。

アクセス許可の付与

S3 テーブルを と統合した後 AWS Lake Formation、S3 テーブルカタログとカタログオブジェクト (テーブルバケット、データベース、テーブル) に対するアクセス許可を、アカウントの他の IAM ロールとユーザーに付与できます。Lake Formation アクセス許可を使用すると、Amazon Redshift Spectrum や Athena などの統合分析エンジンのユーザーに対して、テーブル、列、行レベルの粒度でアクセスコントロールを定義できます。

許可は、名前付きリソース方式、または Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方式を使用して付与することができます。LF タグと LF タグ式を使用してアクセス許可を付与する前に、それらを定義してデータカタログオブジェクトに割り当てる必要があります。

詳細については、「メタデータアクセスコントロールのための LF タグの管理」を参照してください。

外部 AWS アカウントに Lake Formation アクセス許可を付与することで、データベースとテーブルを外部アカウントと共有できます。ユーザーはその後、複数のアカウントにまたがるテーブルを結合してクエリするクエリとジョブを実行できるようになります。データカタログリソースを別のアカウントと共有すると、そのアカウント内のプリンシパルは、そのリソースをプリンシパルのデータカタログ内にあるかのように操作することができます。

データベースとテーブルを外部アカウントと共有すると、スーパーユーザーのアクセス許可は使用できません。

アクセス許可を付与する詳細な手順については、「Lake Formation 許可の管理」セクションを参照してください。

AWS CLI Amazon S3 テーブルに対するアクセス許可の付与例

```
aws lakeformation grant-permissions \
--cli-input-json \
'{
    "Principal": {
        "DataLakePrincipalIdentifier":"arn:aws:iam::111122223333:role/DataAnalystRole"
    },
    "Resource": {
        "Table": {
            "CatalogId":"111122223333:s3tablescatalog/amzn-s3-demo-bucket1",
            "DatabaseName":"S3 table bucket namespace <example_namespace>",
            "Name":"S3 table bucket table name <example_table>"
        }
    },
    "Permissions": [
        "SELECT"
    ]
}'
```

コマンドに含めるパラメータは次のとおりです。

- DataLakePrincipalIdentifier – アクセス許可を付与する IAM ユーザー、ロール、またはグループ ARN
- CatalogId – データカタログを所有する 12 桁の AWS アカウント ID

- DatabaseName – Amazon S3 テーブルバケット名前空間の名前
- Name – Amazon S3 テーブルバケットのテーブル名
- Permissions – 付与するアクセス許可。オプション:
  SELECT、INSERT、DELETE、DESCRIBE、ALTER、DROP、ALLL、SUPER

共有 Amazon S3 Tables へのアクセス

S3 テーブルカタログのデータベースまたはテーブルに対するクロスアカウントアクセス許可を付与した後、リソースにアクセスするには、共有データベースとテーブルへのリソースリンクを作成する必要があります。

1. 送信先アカウント (共有リソースを受信するアカウント) で、データベースリソースリンクを作成します。詳細な手順については、「共有 Data Catalog データベースへのリソースリンクの作成」を参照してください。

   データベースリソースリンクを作成するための CLI の例

   ```
   aws glue create-database
   --region us-east-1
   --catalog-id "111122223333"
   --database-input \
   '{
     "Name": "s3table_resourcelink",
     "TargetDatabase": {
       "CatalogId": "011426214932:s3tablescatalog/chmni-s3-table-bucket-011426214932",
       "DatabaseName": "s3_table_ns"
     },
     "CreateTableDefaultPermissions": []
   }'
   ```

2. テーブルに対するクロスアカウントアクセス許可を付与します。

   クロスアカウントアクセス許可付与の CLI の例

   ```
   aws lakeformation grant-permissions \
   --region us-east-1 \
   --cli-input-json \
   '{
       "Principal": {
           "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:role/
   S3TablesTestExecRole"
   ```

```
    },
    "Resource": {
        "Table": {
            "CatalogId": "011426214932:s3tablescatalog/chmni-s3-table-
bucket-011426214932",
            "DatabaseName": "s3_table_ns",
            "Name": "test_s3_iceberg_table"
        }
    },
    "Permissions": [
        "ALL"
    ]
}'
```

3. リソースリンクに対する Lake Formation DESCRIBE アクセス許可を付与します。

   リソースリンクに対する describe アクセス許可を付与する CLI の例。

```
aws lakeformation grant-permissions \
    --principal DataLakePrincipalIdentifier=arn:aws:iam::111122223333:role/
S3TablesTestExecRole
    --resource Database='{CatalogId=111122223333;, Name=s3table_resourcelink}' \
    --permissions DESCRIBE
```

S3 Tables 統合のアクセスコントロールの変更

Amazon S3 Tables を と統合したら AWS Glue Data Catalog、カタログリソースへのアクセスの制御方法を変更できます。このセクションでは、現在使用しているアクセスコントロールモデルと希望するアクセスコントロールモデルに応じてアクセスコントロールを変更する方法を説明します。Lake Formation を有効にすると、Lake Formation 許可を通じて列レベルや行レベルのセキュリティなどのきめ細かなアクセス許可を使用でき、Lake Formation は登録されたロールを通じてプリンシパルに代わって一時的な認証情報を提供できます。アクセスコントロール AWS Lake Formation を から IAM に変更すると、標準の IAM ポリシーへのアクセスコントロールが返されます。これは、ワークロードにきめ細かなアクセスが必要ではなく、IAM を通じてアクセス許可を完全に管理する場合に適しています。どちらの移行パスにも、データカタログのデフォルトの更新、Lake Formation によるリソース登録の調整、移行中のアクセス中断を回避するためのアクセス許可の付与の調整が含まれます。

トピック

- S3 Tables と Data Catalog の統合で Lake Formation を有効にする
- アクセスコントロール AWS Lake Formation を から IAM に変更する

S3 Tables と Data Catalog の統合で Lake Formation を有効にする

このセクションでは、 と統合された Amazon S3 Tables の AWS Lake Formation 許可を使用して、アクセスコントロールを IAM 権限から IAM に移行するワークフローについて説明します AWS Glue Data Catalog。

> **Important**
>
> AWS Lake Formation アクセスコントロールを有効にすると、S3 Tables リソースへの既存の IAM ベースのアクセスがすべて取り消されます。ステップ 1 を完了すると、IAM アクセス許可を通じて以前にデータにアクセスしたユーザーとロールはすぐにアクセスできなくなります。ユーザーがデータを再度クエリできるようにするには、ステップ 2 で Lake Formation アクセス許可を付与する必要があります。メンテナンス期間中にこの移行を計画し、データチームと調整します。

前提条件

S3 Tables への読み取り/書き込みアクセスには、Lake Formation アクセス許可に加えて、プリンシパルにも `lakeformation:GetDataAccess` IAM アクセス許可が必要です。この許可があると、Lake Formation がデータにアクセスするための一時的な認証情報のリクエストを承諾します。

の使用 AWS CLI

1. ステップ 1: IAM ロールを使用してバケットを Lake Formation に登録する

   S3 Tables リソースを Lake Formation に登録します。

   > **Note**
   >
   > 既存のロールがある場合は、ハイブリッドアクセスが false であることを確認します。

```
aws lakeformation register-resource \
  --resource-arn "arn:aws:s3tables:us-east-1:AWSAccountID:bucket/*" \
```

```
  --role-arn "arn:aws:iam::AWSAccountID:role/service-role/LFAccessRole" \
  --with-federation
```

2. ステップ 2: Lake Formation のアクセスコントロールを有効にするように AWS Glue カタログを更新する

   カタログを空の `CreateDatabaseDefaultPermissions`と `CreateTableDefaultPermissions` (に設定) で更新します`[]`。これにより、カタログ内のデータベースとテーブルは、IAM ベースのデフォルトのアクセス許可ではなく Lake Formation 許可を使用して管理されます。

```
aws glue update-catalog \
  --catalog-id "s3tablescatalog" \
  --catalog-input '{
    "FederatedCatalog": {
        "Identifier": "arn:aws:s3tables:us-east-1:AWSAccountID:bucket/*",
        "ConnectionName": "aws:s3tables"
    },
    "CreateDatabaseDefaultPermissions": [],
    "CreateTableDefaultPermissions": [],
    "AllowFullTableExternalDataAccess": "True"
  }'
```

3. ステップ 3: データチームに Lake Formation アクセス許可を付与する

   アクセスを必要とするプリンシパル (ロール、ユーザー、またはグループ) に Lake Formation のアクセス許可を付与します。たとえば、フルテーブルの読み取りアクセスをロールに付与するには、次のようにします。

```
aws lakeformation grant-permissions \
  --principal
 DataLakePrincipalIdentifier=arn:aws:iam::AWSAccountID:role/DataTeamRole \
  --resource '{
    "Table": {
        "CatalogId": "AWSAccountID",
        "DatabaseName": "s3tablescatalog/table-bucket-name/namespace",
        "TableWildcard": {}
    }
  }' \
  --permissions "SELECT" "DESCRIBE"
```

   必要に応じて、プリンシパルとリソースの組み合わせごとに繰り返します。

アクセスコントロール AWS Lake Formation を から IAM に変更する

このセクションでは、 と統合された Amazon S3 Tables のアクセスコントロールを AWS Lake Formation 権限から IAM 権限に変更するワークフローについて説明します AWS Glue Data Catalog。

> **Important**
>
> アクセスコントロールを AWS Lake Formation 権限から IAM に変更すると、S3 Tables リソースへの既存の Lake Formation ベースのアクセスはすべて取り消されます。ステップ 2 を完了すると、以前に Lake Formation 許可を通じてデータにアクセスしたユーザーとロールはすぐにアクセスできなくなります。カタログを更新する前に、ステップ 1 で IAM アクセスを許可する必要があります。メンテナンス期間中にこの移行を計画し、データチームと調整します。

> **Important**
>
> Data Catalog オブジェクトを使用した列レベルのアクセスやデータセルフィルターなどのきめ細かなアクセスコントロールは、 AWS Lake Formation を使用する場合にのみ使用できます。アクセスコントロールを から IAM AWS Lake Formation に移行する前に、 を使用して既存の Lake Formation 許可を監査`aws lakeformation list-permissions`し、同等の IAM ポリシーがユーザーに必要なアクセスを提供できるかどうかを確認します。Lake Formation のきめ細かな許可に依存していたプリンシパルには、アクセスコントロールの移行後にテーブルレベルのフルアクセスが必要です。

前提条件

開始する前に、以下を確認してください。

- 移行するリソースに対して現在有効なすべての Lake Formation 許可を特定しました。`aws lakeformation list-permissions --resource-type TABLE` を実行して確認します。
- 影響を受けるすべてのプリンシパルに同等のアクセスを提供する IAM ポリシーを準備しました。
- Lake Formation に登録された IAM ロールはまだ です `lakeformation:GetDataAccess` (ハイブリッド移行期間中に必要) 。

の使用 AWS CLI

1. ステップ 1: プリンシパルに IAM アクセス許可を付与する

   アクセスが必要なユーザーまたはロールに IAM ポリシーをアタッチします。ポリシーには、AWS Glue メタデータアクセス許可と S3 Tables データアクセス許可の両方を含める必要があります。

   > **Note**
   >
   > 次のポリシー例では、読み取りアクセスのみを提供します。

```
aws iam put-user-policy \
  --user-name GlueIAMAccessUser \
  --policy-name S3TablesIAMAccessPolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GlueMetadataAccess",
            "Effect": "Allow",
            "Action": [
                "glue:GetCatalog",
                "glue:GetDatabase",
                "glue:GetTable"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:AWSAccountID:catalog/s3tablescatalog",
                "arn:aws:glue:us-east-1:AWSAccountID:database/
s3tablescatalog/table-bucket-name/namespace",
                "arn:aws:glue:us-east-1:AWSAccountID:table/s3tablescatalog/table-
bucket-name/namespace/*"
            ]
        },
        {
            "Sid": "S3TablesDataAccess",
            "Effect": "Allow",
            "Action": [
                "s3tables:GetTableBucket",
                "s3tables:GetTable",
                "s3tables:GetTableMetadataLocation",
```

```
                "s3tables:GetTableData"
            ],
            "Resource": [
                "arn:aws:s3tables:us-east-1:AWSAccountID:bucket/table-bucket-name",
                "arn:aws:s3tables:us-east-1:AWSAccountID:bucket/table-bucket-name/
table/*"
            ]
        }
    ]
  }'
```

先に進む前に、影響を受けるすべてのユーザーとロールが、IAM 認証情報を使用して期待されるテーブルにアクセスできることを確認します。

2. ステップ 2: カタログを更新して IAM のデフォルトのアクセス許可を復元する

   カタログを更新して、 CreateDatabaseDefaultPermissionsと を ALLにCreateTableDefaultPermissions付与しますIAM_ALLOWED_PRINCIPALS。を OverwriteChildResourcePermissionsWithDefaultに設定するAcceptと、変更は新しく作成された子リソースだけでなく、既存のすべての子リソースに反映されます。

```
aws glue update-catalog \
  --catalog-id "s3tablescatalog" \
  --catalog-input '{
    "FederatedCatalog": {
        "Identifier": "arn:aws:s3tables:us-east-1:AWSAccountID:bucket/*",
        "ConnectionName": "aws:s3tables"
    },
    "CreateDatabaseDefaultPermissions": [{
        "Principal": {"DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"},
        "Permissions": ["ALL"]
    }],
    "CreateTableDefaultPermissions": [{
        "Principal": {"DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"},
        "Permissions": ["ALL"]
    }],
    "OverwriteChildResourcePermissionsWithDefault": "Accept"
  }'
```

3. ステップ 3: Lake Formation からリソースの登録を解除する

すべてのアクセスが IAM ポリシーを通じて機能し、プリンシパルが Lake Formation の許可に依存していないことを確認したら、Lake Formation からリソースを登録解除して移行を完了できます。

```
aws lakeformation deregister-resource \
  --resource-arn "arn:aws:s3tables:us-east-1:AWSAccountID:bucket/*"
```

> **Note**
>
> リソースの登録を解除したら、不要になった IAM プリンシパルlakeformation:GetDataAccessから を削除します。

revoke-permissions ステップは必要ありません。

での Amazon Redshift マネージドカタログの作成 AWS Glue Data Catalog

現在 Amazon Redshift プロデューサークラスターや Amazon Redshift データ共有は利用できない場合がありますが、 AWS Glue Data Catalogを使用して Amazon Redshift テーブルを作成および管理する必要があります。glue:CreateCatalog API または AWS Lake Formation コンソールを使用して AWS Glue マネージドカタログを作成するには、カタログタイプを Catalog source Redshift として設定Managedします。 この手順では、以下を実行します。

- データカタログに新しいカタログを作成する
- カタログを Lake Formation データの場所として登録する
- Amazon Redshift マネージドサーバーレスワークグループを作成する
- データ共有オブジェクトを使用して Amazon Redshift Serverless ワークグループとデータカタログをリンクする

マネージドカタログを作成し、アクセス許可を設定するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。
2. ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。
3. [カタログの作成] オプションを選択します。

4. [カタログの設定] の詳細ページで、次の情報を入力します。

   - 名前 – マネージドカタログの一意の名前。名前は変更できず、小文字にする必要があります。名前は最大 255 文字で、アカウント内で一意である必要があります。
   - タイプ – カタログタイプとして `Managed catalog` を選択します。
   - ストレージ – ストレージとして `Redshift` を選択します。
   - 説明 – データソースから作成されたカタログの説明を入力します。

5. Amazon EC2 での Amazon EMR で実行されている Apache Spark アプリケーションを使用して、 AWS Glue Data Catalogの Amazon Redshift データベースにアクセスできます。

   Apache Spark による Amazon Redshift マネージドストレージの読み取りや書き込みができるように、 AWS Glue は Amazon Redshift データウェアハウスのワークロードに影響を与えずに読み取り操作や書き込み操作を実行するために必要なコンピューティングリソースとストレージリソースを備えたマネージド Amazon Redshift クラスターを作成します。Amazon S3 バケットとの間でデータを転送するために必要なアクセス許可を持つ IAM ロールも指定する必要があります。データ転送ロールに必要なアクセス許可については、「で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog」セクションの「ステップ 5」を参照してください。

6. デフォルトでは、Amazon Redshift クラスター内のデータは AWS マネージドキーを使用して暗号化されます。Lake Formation には、暗号化用のカスタム KMS キーを作成するオプションがあります。カスタマーマネージドキーを使用している場合は、特定のキーポリシーをキーに追加する必要があります。

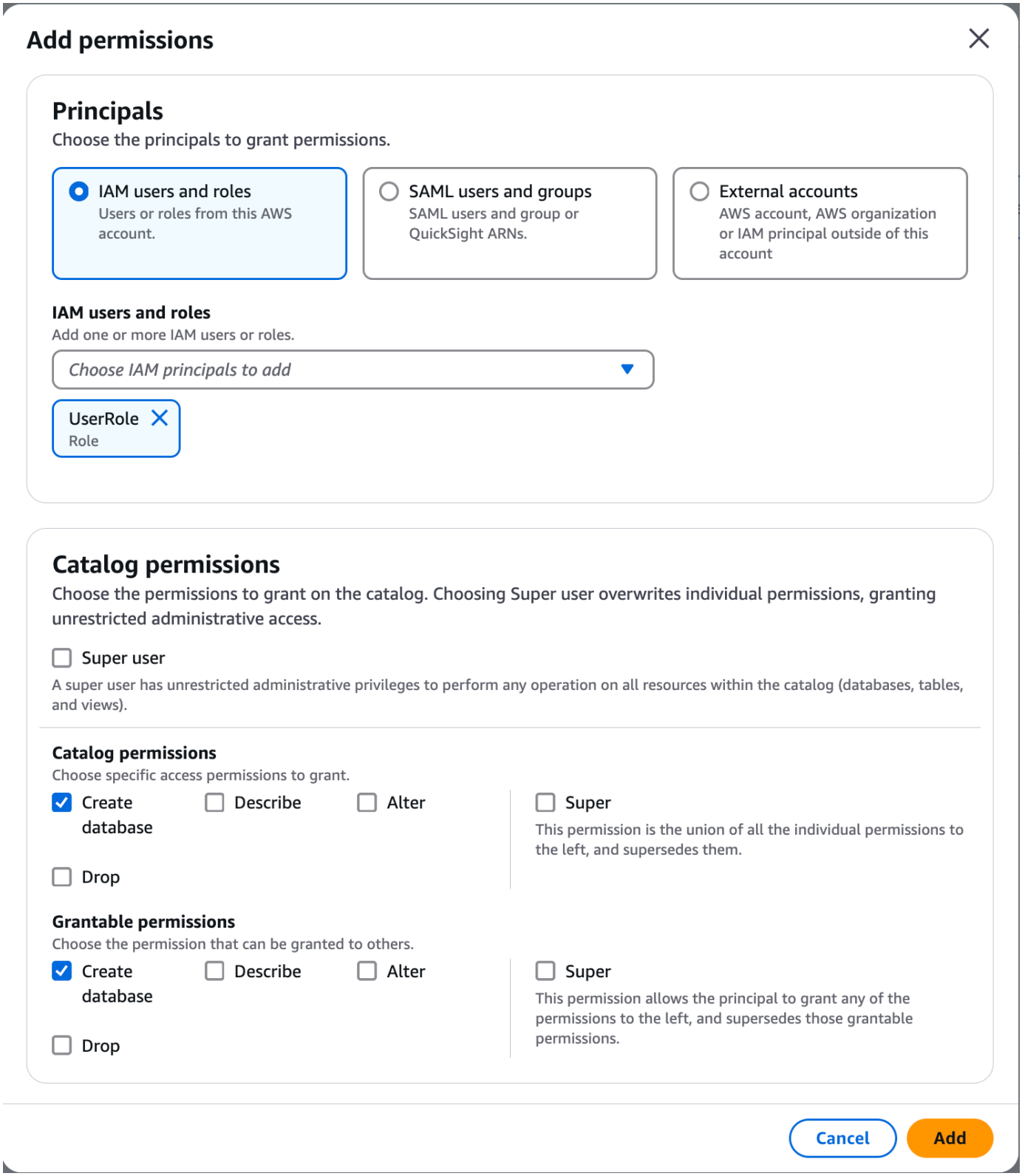
7. カスタマーマネージドキーを使用して Amazon Redshift マネージドストレージ内のデータを暗号化する場合は、[暗号化設定のカスタマイズ] を選択します。カスタムキーを使用するには、KMS キーにカスタムマネージドキーポリシーを追加する必要があります。詳細については、「で Amazon Redshift 名前空間を管理するための前提条件 AWS Glue Data Catalog」を参照してください。

8. 暗号化オプション – カスタムキーを使用してカタログを暗号化する場合は、[暗号化設定のカスタマイズ] オプションを選択します。カスタムキーを使用するには、KMS キーにカスタムマネージドキーポリシーを追加する必要があります。

9. [次へ] を選択して、他のプリンシパルにアクセス許可を付与します。

10. [アクセス許可を付与] ページで、[アクセス許可を追加] を選択します。

11. [アクセス許可を追加] 画面で、プリンシパルと、付与するアクセス許可のタイプを選択します。



- [Principals] (プリンシパル) セクションでプリンシパルタイプを選択してから、アクセス許可の付与先となるプリンシパルを指定します。

- IAM ユーザーとロール – IAM ユーザーとロールのリストから、ユーザーまたはロールを 1 つ以上選択します。
- SAML ユーザーとグループ – SAML および Amazon Quick ユーザーとグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、または Amazon Quick ユーザーまたはグループの ARNs を入力します。各 ARN の後で [Enter] キーを押します。

  ARNs」を参照してください。 AWS CLI AWS CLI

- [Permissions] (許可) セクションで、許可と付与可能な許可を選択します。

  [カタログのアクセス許可] で、付与するアクセス許可を 1 つ以上選択します。

  カタログ内のすべてのリソースに対する無制限の管理アクセス許可を付与するには、[スーパーユーザー] を選択します。

  付与可能なアクセス許可で、付与受信者が AWS アカウントの他のプリンシパルに付与できるアクセス許可を選択します。このオプションは、外部アカウントから IAM プリンシパルにアクセス許可を付与する場合はサポートされません。

12. [次へ] を選択して情報を確認し、カタログを作成します。[カタログ] リストには、新しいマネージドカタログが表示されます。

フェデレーティッドカタログを作成するには (CLI)

- 次の例は、フェデレーティッドカタログを作成する方法を示しています。

```
aws glue create-catalog --cli-input-json file://input.json

{
    "Name": "CatalogName",
    "CatalogInput": {
      "Description": "Redshift published Catalog",
      "CreateDatabaseDefaultPermissions" : [],
      "CreateTableDefaultPermissions": [],
      "CatalogProperties": {
        "DataLakeAccessProperties" : {
          "DataLakeAccess" : "true",
          "DataTransferRole" : "DTR arn",
          "KMSKey": "kms key arn",  // Optional
          "CatalogType": "aws:redshift"
```

```
        }
      }
    }
}
```

Glue get-catalog レスポンス

```
aws glue get-catalog \
  --catalog-id account-id:catalog-name \
  --region us-east-1

Response:
{
    "Catalog": {
        "Name": "CatalogName",
        "Description": "Glue Catalog for Redshift z-etl use case",
        "CreateDatabaseDefaultPermissions" : [],
        "CreateTableDefaultPermissions": [],
         "CatalogProperties": {
          "DataLakeAccessProperties" : {
            "DataLakeAccess": "true",
            "DataTransferRole": "DTR arn",
            "KMSKey": "kms key arn",
            "ManagedWorkgroupName": "MWG name",
            "ManagedWorkgroupStatus": "MWG status",
            "RedshiftDatabaseName": "RS db name",
            "NamespaceArn": "namespace key arn",
            "CatalogType": "aws:redshift"
         }
       }
    }
```

Amazon Redshift データ共有でのデータに対するアクセス許可の管理

AWS Lake Formation では、Amazon Redshift のデータ共有のデータを安全に管理できます。Amazon Redshift は、AWS クラウド内でのフルマネージド型、ペタバイト規模のデータウェアハウスサービスです。Amazon Redshift では、データ共有機能を使用して、AWS アカウント 間で

データを共有できます。Amazon Redshift データ共有の詳細については、「Amazon Redshift でのデータ共有の概要」を参照してください。

Amazon Redshift では、プロデューサークラスター管理者がデータ共有を作成し、データレイク管理者と共有します。データレイク管理者を作成するための詳しい手順については、「データレイク管理者を作成する」を参照してください。

ユーザー (データレイク管理者) がデータ共有を承諾したら、特定のデータ共有用の AWS Glue Data Catalog データベースを作成する必要があります。これは、Lake Formation のアクセス許可を使用してアクセスを制御できるようにするためです。Lake Formation は、各データ共有を対応するデータカタログデータベースにマッピングします。これらはデータカタログにフェデレーションデータベースとして表示されます。

データカタログ外のエンティティを指すデータベースは、フェデレーションデータベースと呼ばれます。Amazon Redshift データ共有のテーブルとビューは、データカタログに個別のテーブルとして表示されます。フェデレーションデータベースは、同じアカウントまたは Lake Formation の別のアカウント内の、選択した IAM プリンシパルおよび SAML ユーザーと共有できます。行と列のフィルター式を含めて、特定データへのアクセスを制限することもできます。詳細については、「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」を参照してください。

ユーザーに Amazon Redshift データ共有へのアクセスを提供するには、以下の操作を実行する必要があります。

1. [Data Catalog settings] (データカタログの設定) を更新して、Lake Formation アクセス許可を有効にします。
2. Amazon Redshift プロデューサークラスター管理者からのデータ共有の招待を承諾し、データ共有を Lake Formation に登録します。

   この手順を完了すると、Lake Formation データカタログ内でデータ共有を管理できるようになります。
3. フェデレーションデータベースを作成し、そのデータベースに対するアクセス許可を定義します。
4. データベースとテーブルに対するアクセス許可をユーザーに付与します。データベース全体またはテーブルのサブセットを、同じアカウント内または別のアカウント内のユーザーと共有できます。

制限事項については、「Amazon Redshift データ共有の制限事項」を参照してください。

トピック

- Amazon Redshift データ共有に対するアクセス許可設定の前提条件
- Amazon Redshift データ共有に対するアクセス許可の設定
- フェデレーションデータベースのクエリ

Amazon Redshift データ共有に対するアクセス許可設定の前提条件

デフォルトのデータカタログ設定を更新します

データカタログリソースの Lake Formation アクセス許可を有効にするには、Lake Formation のデフォルトの [Data Catalog settings] (データカタログの設定) を無効にすることをお勧めします。詳細については、「デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する」を参照してください。

アクセス許可の更新

Lake Formation で Amazon Redshift データ共有を承諾するには、データレイク管理者アクセス許可 (`AWSLakeFormationDataAdmin`) に加えて、以下のアクセス許可も必要です。

- `glue:PassConnection on aws:redshift`
- `redshift:AssociateDataShareConsumer`
- `redshift:DescribeDataSharesForConsumer`
- `redshift:DescribeDataShares`

データレイク管理者 IAM ユーザーには、以下のアクセス許可が暗黙的に付与されます。

- data_location_access
- create_database
- lakefomation:registerResource

Amazon Redshift データ共有に対するアクセス許可の設定

このトピックでは、データ共有への招待を承諾し、フェデレーションデータベースを作成し、アクセス許可を付与するために必要なステップについて説明します。Lake Formation コンソールまたは AWS Command Line Interface (AWS CLI) を使用できます。このトピックの例では、同じアカウントのプロデューサークラスター、データカタログ、およびデータコンシューマーを示しています。

Lake Formation のクロスアカウント機能の詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

データ共有にアクセス許可を設定するには

1. データ共有への招待を確認して承諾します。

   Console

   1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にデータレイク管理者としてサインインします。[Data sharing] (データ共有) ページに移動します。
   2. アクセスが許可されているデータ共有を確認します。[Status] (ステータス) 列は、データ共有の現在の参加ステータスを示します。[Pending] (保留中) ステータスは、ユーザーがデータ共有に追加されたが、招待を承諾または拒否していないことを示します。
   3. データ共有の招待に応答するには、データ共有名を選択し、[招待を確認] を選択します。[データ共有の承諾または拒否] で、招待の詳細を確認します。[Accept] (承諾) を選択して招待を承諾するか、[Reject] (拒否) を選択して招待を却下します。招待を拒否した場合、データ共有にはアクセスできません。

   AWS CLI

   以下の例では、招待を表示、承諾、登録する方法を示します。AWS アカウント ID は、有効な AWS アカウント ID に置き換えてください。`data-share-arn` を、データ共有を参照する実際の Amazon リソースネーム (ARN) に置き換えます。

   1. 保留中の招待を確認します。

      ```
      aws redshift describe-data-shares \
       --data-share-arn 'arn:aws:redshift:us-
      east-1:111122223333:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/
      federatedds' \
      ```

   2. データ共有の承諾

      ```
       aws redshift associate-data-share-consumer \
       --data-share-arn 'arn:aws:redshift:us-
      east-1:111122223333:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/
      federatedds' \
       --consumer-arn 'arn:aws:glue:us-east-1:111122223333:catalog
      ```

3. Lake Formation アカウントにデータ共有を登録します。RegisterResource API オペレーションを使用して、データ共有を Lake Formation に登録します。DataShareArn は ResourceArn の入力パラメータです。

> **Note**
>
> これは必須の手順です。

```
aws lakeformation register-resource \
 --resource-arn 'arn:aws:redshift:us-
east-1:111122223333:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/
federatedds'
```

2. データベースを作成します。

データ共有の招待を承諾したら、データ共有に関連付けられた Amazon Redshift データベースを指すデータベースを作成する必要があります。データベースを作成するには、データレイク管理者である必要があります。

Console

1. [Invitations] (招待) ペインからデータ共有を選択し、[Set database details] (データベース詳細の設定) を選択します。
2. [Set database details] (データベース詳細の設定) に、データ共有の固有の名前と ID を入力します。この ID は、メタデータ階層 (dbName.schema.table) でデータ共有を内部的にマッピングするために使用されます。
3. 共有データベースとテーブルに対するアクセス許可を他のユーザーに付与するには、[Next] (次へ) を選択します。

AWS CLI

次のサンプルコードを使用して、Lake Formation と共有されている Amazon Redshift データベースを指すデータベースを AWS CLI で作成します。

```
aws glue create-database --cli-input-json \

'{
```

```
 "CatalogId": "111122223333",
 "DatabaseInput": {
  "Name": "tahoedb",
  "FederatedDatabase": {
       "Identifier": "arn:aws:redshift:us-
east-1:111122223333:datashare:abcd1234-1234-ab12-cd34-1a2b3c4d5e6f/federatedds",
       "ConnectionName": "aws:redshift"
   }
 }
 }'
```

3. アクセス許可を付与します。

データベースを作成したら、アカウント内の他のユーザーや外部の AWS アカウントと組織にアクセス許可を付与できます。Amazon Redshift データ共有にマッピングされたフェデレーションデータベースには、データの書き込みアクセス許可 (挿入、削除) とメタデータのアクセス許可 (変更、ドロップ、作成) を付与することはできません。許可の付与の詳細については、「Lake Formation 許可の管理」を参照してください。

> **Note**
>
> データレイク管理者は、フェデレーションデータベース内のテーブルの表示のみを行うことができます。他のアクションを実行するには、それらのテーブルに対する他のアクセス許可を付与する必要があります。

Console

1. [Grant permissions] (アクセス許可の付与) 画面で、アクセス許可を付与するユーザーを選択します。
2. [Grant] (付与) を選択します。

AWS CLI

以下の例を使用して、データベースとテーブルのアクセス許可を AWS CLI で付与します。

```
aws lakeformation grant-permissions --input-cli-json file://input.json

{
   "Principal": {
```

```
           "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/non-
admin"
   },
   "Resource": {
          "Database": {
                "CatalogId": "111122223333",
                 "Name": "tahoedb"
           }
    },
    "Permissions": [
             "DESCRIBE"
     ],
    "PermissionsWithGrantOption": [

     ]
 }
```

```
aws lakeformation grant-permissions --input-cli-json file://input.json

{
                   "Principal": {
                          "DataLakePrincipalIdentifier":
 "arn:aws:iam::111122223333:user/non-admin"
                   },
                  "Resource": {
                         "Table": {
                              "CatalogId": "111122223333",
                              "DatabaseName": "tahoedb",
                              "Name": "public.customer"
                         }
                  },
                  "Permissions": [
                        "SELECT"
                  ],
                  "PermissionsWithGrantOption": [
                         "SELECT"
                   ]
 }
```

フェデレーションデータベースのクエリ

アクセス許可の付与後、ユーザーは Amazon Redshift を使用してサインインし、フェデレーションデータベースへのクエリを開始できます。これで、ユーザーはローカルデータベース名を使用して SQL クエリで Amazon Redshift データ共有を参照できるようになります。Amazon Redshift では、データ共有を介して共有されるパブリックスキーマの顧客テーブルには、データカタログの `public.customer` として作成される、対応するテーブルが作成されます。

1. Amazon Redshift を使用してフェデレーションデータベースにクエリを実行する前に、クラスター管理者は次のコマンドを使用してデータカタログデータベースからデータベースを作成します。

```
CREATE DATABASE sharedcustomerdb FROM ARN
 'arn:aws:glue:<region>:111122223333:database/tahoedb' WITH DATA CATALOG SCHEMA
 tahoedb
```

2. クラスター管理者は、データベースでの使用に関するアクセス許可を付与します。

```
GRANT USAGE ON DATABASE sharedcustomerdb TO IAM:user;
```

3. これで、フェデレーションユーザーは SQL ツールにログインしてテーブルをクエリできます。

```
Select * from sharedcustomerdb.public.customer limit 10;
```

詳細については、「Amazon Redshift 管理ガイド」の「AWS Glue Data Catalog のクエリ」を参照してください。

外部メタストアを使用するデータセットのアクセス許可の管理

AWS Glue Data Catalog メタデータフェデレーション (データカタログフェデレーション) を使用すると、データカタログを Amazon S3 データのメタデータを保存する外部メタストアに接続し、 を使用してデータアクセス許可を安全に管理できます AWS Lake Formation。メタデータを外部メタストアからデータカタログに移行する必要はありません。

データカタログは、一元化されたメタデータリポジトリを提供し、異種システム間でのデータの管理と発見を容易にします。組織がデータカタログ内のデータを管理する場合、 AWS Lake Formation を使用して Amazon S3 内のデータセットへのアクセスを制御できます。

> **Note**
>
> 現在、Apache Hive (バージョン 3 以降) メタストアフェデレーションのみをサポートしています。

データカタログフェデレーションを設定するには、 で GlueDataCatalogFederation-HiveMetastore と呼ばれる AWS Serverless Application Model (AWS SAM) アプリケーションを提供します AWS Serverless Application Repository。

リファレンス実装は、AWS Glue Data Catalog フェデレーション - Hive メタストアのオープンソースプロジェクトとして GitHub で提供されています。

AWS SAM アプリケーションは、データカタログを Hive メタストアに接続するために必要な次のリソースを作成してデプロイします。

- AWS Lambda 関数 – データカタログと Hive metastore の間で通信するフェデレーションサービスの実装をホストします。 はこの Lambda 関数を AWS Glue 呼び出して Hive メタストアからメタデータオブジェクトを取得します。
- Amazon API Gateway – すべての呼び出しを Lambda 関数にルーティングするプロキシとして機能する Hive メタストアの接続エンドポイント。
- IAM ロール - データカタログと Hive メタストア間の接続を作成するために必要なアクセス許可を持つロール。
- AWS Glue connection – Amazon API Gateway エンドポイントと、エンドポイントを呼び出す IAM ロールを保存する AWS Glue 接続 Amazon API Gateway のタイプ。

テーブルをクエリすると、 AWS Glue サービスは Hive メタストアへのランタイム呼び出しを行い、メタデータを取得します。Lambda 関数は、Hive メタストアとデータカタログ間のトランスレータとして機能します。

接続を確立した後、Hive メタストアのメタデータをデータカタログと同期するために、Hive メタストア接続の詳細を使用してデータカタログにフェデレーションデータベースを作成し、このデータベースを Hive データベースにマッピングする必要があります。データベースは、データカタログ外のエンティティを指す場合、フェデレーションデータベースと呼ばれます。

タグベースのアクセスコントロールと名前付きリソースメソッドを使用して Lake Formation のアクセス許可をフェデレーションデータベースに適用し、複数 AWS アカウント、 AWS Organizations、

および組織単位 (OUs) 間で共有できます。フェデレーションデータベースは、別のアカウントの IAM プリンシパルと直接共有することもできます。

外部 Hive テーブルで Lake Formation データフィルターを使用すると、列レベル、行レベル、およびセルレベルできめ細かいアクセス許可を定義できます。Amazon Athena、Amazon Redshift、または Amazon EMR を使用して、Lake Formation が管理する外部 Hive テーブルにクエリを実行できます。

クロスアカウントデータ共有およびデータフィルタリングの詳細については、以下を参照してください。

- Lake Formation でのクロスアカウントデータ共有
- Lake Formation でのデータフィルタリングとセルレベルのセキュリティ

データカタログメタデータフェデレーションの手順の概要

1. AWS SAM アプリケーションをデプロイし、フェデレーションデータベースを作成するための適切なアクセス許可を持つ IAM ユーザーとロールを作成します。
2. 外部 Hive メタストアを使用するデータセットの `Enable Data Catalog federation` オプションを選択して、Amazon S3 のデータロケーションを Lake Formation に登録します。
3. AWS SAM アプリケーション設定 (AWS Glue 接続名、Hive メタストアへの URL、Lambda 関数パラメータ) を設定し、 AWS SAM アプリケーションをデプロイします。
4. AWS SAM アプリケーションは、外部 Hive メタストアを Data Catalog に接続するために必要なリソースをデプロイします。
5. Hive データベースとテーブルに Lake Formation アクセス許可を適用するには、Hive メタストア接続の詳細を使用してデータカタログにデータベースを作成し、このデータベースを Hive データベースにマッピングします。
6. フェデレーションデータベースのアクセス許可を、自分のアカウントまたは別のアカウントのプリンシパルに付与します。

> **Note**
>
> Lake Formation のアクセス許可を適用しなくても、データカタログを外部 Hive メスタストアに接続したり、フェデレーションデータベースを作成したり、Hive データベースやテーブルでクエリや ETL スクリプトを実行したりできます。Lake Formation に登録されていない

Amazon S3 のソースデータの場合、アクセスは Amazon S3 の IAM アクセス許可ポリシーと AWS Glue アクションによって決まります。

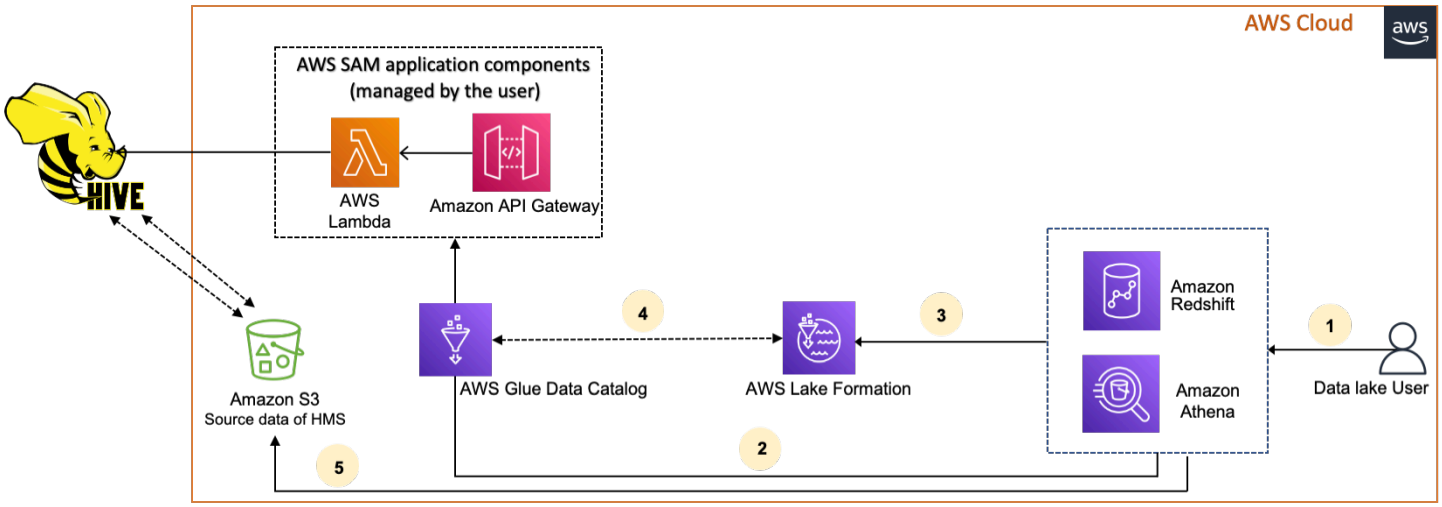
制限事項については、「Hive メタデータストアのデータ共有に関する考慮事項と制限事項」を参照してください。

トピック

- ワークフロー
- データカタログを Hive メタストアに接続するための前提条件
- データカタログを外部 Hive メタストアに接続する
- その他のリソース

ワークフロー

次の図は、 AWS Glue Data Catalog を外部 Hive メタストアに接続するためのワークフローを示しています。



1. プリンシパルは、Athena や Redshift Spectrum などの統合サービスを使用してクエリを送信します。
2. 統合サービスは、メタデータの Data Catalog を呼び出します。これにより、その背後で使用可能な Hive メタストアエンドポイントが呼び出され Amazon API Gateway、メタデータリクエストへの応答を受け取ります。
3. 統合サービスが Lake Formation にリクエストを送信し、テーブル情報とテーブルにアクセスするための認証情報を検証します。

4. Lake Formation はリクエストを承認し、統合アプリケーションに一時的な認証情報を提供して、データアクセスを許可します。
5. 統合サービスが Lake Formation から受け取った一時的な認証情報を使用して Amazon S3 からデータを読み取り、結果をプリンシパルと共有します。

データカタログを Hive メタストアに接続するための前提条件

AWS Glue Data Catalog を外部の Apache Hive メタストアに接続してデータアクセス許可を設定するには、次の要件を満たす必要があります。

> **Note**
>
> Lake Formation 管理者が AWS SAM アプリケーションをデプロイし、特権ユーザーのみが Hive メタストア接続を使用して対応するフェデレーティッドデータベースを作成することをお勧めします。

1. IAM ロールを作成します。

   AWS SAM アプリケーションをデプロイするには

   - Hive メタストアへの接続の作成に必要なリソース (Lambda 関数、 Amazon API Gateway、IAM ロール、および AWS Glue 接続) をデプロイするために必要なアクセス許可を持つロールを作成します。

   フェデレーションデータベースを作成するには

   リソースに以下のアクセス許可が必要です。

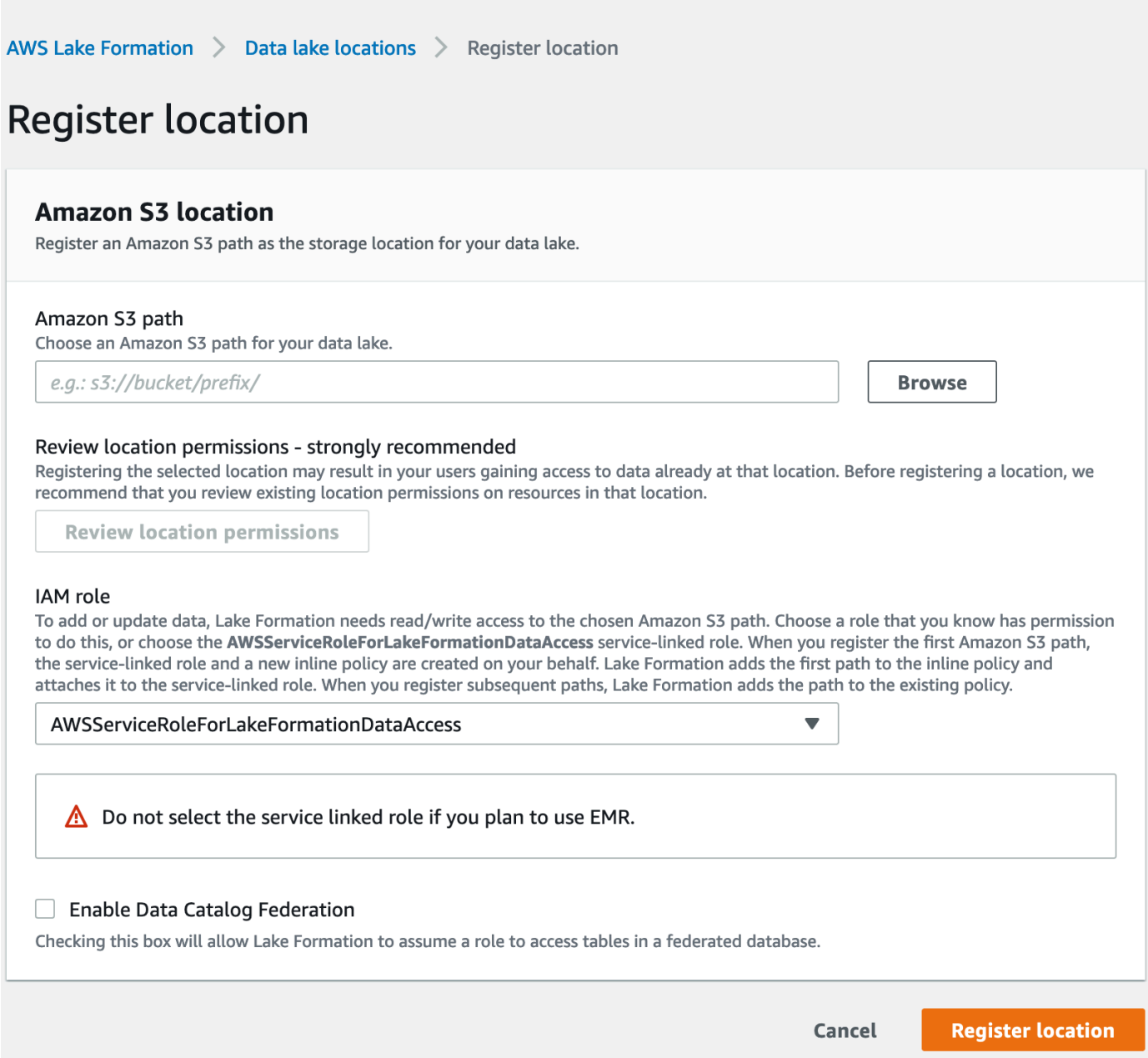
   - `glue:CreateDatabase on resource arn:aws:glue:region:account-id:database/gluedatabasename`
   - `glue:PassConnection on resource arn:aws:glue:region:account-id:connection/hms_connection`
2. Amazon S3 ロケーションを Lake Formation に登録します。

   Lake Formation を使用してデータレイク内のデータを管理および保護するには、Hive メタストアのテーブルのデータを含む Amazon S3 ロケーションを Lake Formation に登録する必要が

あります。これにより、Lake Formation は Athena、Redshift Spectrum、Amazon EMR などの AWS 分析サービスに認証情報を提供できます。

Amazon S3 ロケーションの登録の詳細については「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

Amazon S3 ロケーションを登録するときに、[データカタログフェデレーションを有効にする] チェックボックスをオンにすると、Lake Formation がフェデレーションデータベースのテーブルにアクセスするロールを引き受けることができます。



データロケーションに Lake Formation を登録する方法の詳細については、「データレイク用の Amazon S3 ロケーションを設定する」を参照してください。

3. 適切な Amazon EMR バージョンを使用します。

   Amazon EMR をフェデレーション Hive メタストアデータベースと共に使用するには、Hive バージョン 3.x 以降と Amazon EMR バージョン 6.x 以降が必要です。

データカタログを外部 Hive メタストアに接続する

AWS Glue Data Catalog を Hive メタストアに接続するには、GlueDataCatalogFederation-HiveMetastore という AWS SAM アプリケーションをデプロイする必要があります。外部 Hive メタストアをデータカタログに接続するために必要なリソースを作成します。で AWS SAM アプリケーションにアクセスできます AWS Serverless Application Repository。

AWS SAM アプリケーションは、Lambda 関数を使用して Amazon API Gateway の背後にある Hive メタストアの接続を作成します。 AWS SAM アプリケーションは、ユーザーからの入力としてユニフォームリソース識別子 (URI) を使用し、外部の Hive メタストアをデータカタログに接続します。ユーザーが Hive テーブルでクエリを実行すると、データカタログは API ゲートウェイエンドポイントを呼び出します。エンドポイントは Lambda 関数を呼び出して、Hive テーブルのメタデータを取得します。

データカタログを Hive メタストアに接続してアクセス許可を設定するには

1. AWS SAM アプリケーションをデプロイします。

   1. にサインイン AWS マネジメントコンソール し、 を開きます AWS Serverless Application Repository。
   2. ナビゲーションペインで、[Available applications] (利用可能なアプリケーション) を選択します。
   3. [パブリックアプリケーション] を選択します。
   4. [Show apps that create custom roles or resource policies] (カスタム IAM ロールまたはリソースポリシーを作成するアプリを表示する) オプションを選択します。
   5. 検索ボックスに、[GlueDataCatalogFederation-HiveMetastore] という名前を入力します。
   6. [GlueDataCatalogFederation-HiveMetastore] アプリケーションを選択します。
   7. [アプリケーション設定] で、Lambda 関数に最低限必要な次の設定を入力します。
      - アプリケーション名 - AWS SAM アプリケーションの名前。

- [GlueConnectionName] - 接続の名前。
- [HiveMetastoreURIs] - Hive メタストアホストの URI。
- [LambdaMemory] - Lambda メモリ量 (MB 単位)。128 ~ 10240。デフォルトは 1024 です。
- [LambdaTimeout] - Lambda 呼び出しの最大ランタイム (秒単位)。デフォルトは 30 です。
- [VPCSecurityGroupIds] と [VPCSubnetIds] - Hive メタストアが存在する VPC の情報。

8. [I acknowledge that this app creates custom IAM roles and resource policies] (このアプリがカスタム IAM ロールとリソースポリシーを作成することを承認します) を選択します。詳細については、[Info] (情報) リンクを選択してください。
9. アプリケーション設定セクションで、デプロイを選択します。デプロイが完了すると、Lambda 関数が Lambda コンソールの [リソース] セクションに表示されます。

アプリケーションは Lambda にデプロイされます。その名前の先頭には、アプリケーションが AWS Serverless Application Repositoryからデプロイされたことを示す serverlessrepo- が付けられます。アプリケーションを選択すると、デプロイされたアプリケーションの各リソースが一覧表示される [リソース] ページに移動します。リソースには、データカタログと Hive メタストア間の通信を許可する Lambda 関数、 AWS Glue 接続、およびデータベースフェデレーションに必要なその他のリソースが含まれます。

2. データカタログでフェデレーションデータベースを作成する

Hive メタストアへの接続を作成したら、外部 Hive メタストアデータベースを指すフェデレーションデータベースをデータカタログ内に作成できます。データカタログに接続する Hive メタストアデータベースごとに、対応するデータベースをデータカタログ内に作成する必要があります。

Lake Formation console

1. [データ共有] ページで、[共有データベース] タブを選択し、[データベースの作成] を選択します。
2. [接続名] で、ドロップダウンメニューから Hive メタストア接続の名前を選択します。
3. 一意のデータベース名とデータベースのフェデレーションソース識別子を入力します。これは、テーブルをクエリするときに SQL ステートメントで使用する名前です。名前は最大 255 文字で、アカウント内で一意である必要があります。
4. [データベースの作成] を選択します。

AWS CLI

```
aws glue create-database \
'{
 "CatalogId": "<111122223333>",
  "database-input": {
    "Name":"<fed_glue_db>",
    "FederatedDatabase":{
        "Identifier":"<hive_db_on_emr>",
        "ConnectionName":"<hms_connection>"
     }
   }
 }'
```

3. フェデレーションデータベース内のテーブルを表示します。

フェデレーションデータベースを作成したら、Lake Formation コンソールまたは AWS CLIを使用して Hive メタストア内のテーブルのリストを表示できます。

Lake Formation console

1. [共有データベース] タブからデータベース名を選択します。
2. [データベース] ページで、[テーブルの表示] を選択します。

AWS CLI

次の例は、接続定義、データベース名、データベース内の一部またはすべてのテーブルを取得する方法を示しています。データカタログの ID を、データベースの作成に使用した有効な AWS アカウント ID に置き換えます。hms_connection を接続名に置き換えます。

```
aws glue get-connection \
--name <hms_connection>  \
--catalog-id 111122223333
```

```
aws glue get-database \
--name <fed_glu_db> \
--catalog-id 111122223333
```

```
aws glue get-tables \
--database-name <fed_glue_db> \
--catalog-id 111122223333
```

```
aws glue get-table \
--database-name <fed_glue_db> \
--name <hive_table_name> \
--catalog-id 111122223333
```

4. アクセス許可を付与します。

   データベースを作成したら、アカウントの他の IAM ユーザーとロール、または外部 AWS アカウント と組織にアクセス許可を付与できます。フェデレーションデータベースには、データの書き込みアクセス許可 (挿入、削除) とメタデータのアクセス許可 (変更、ドロップ、作成) を付与することはできません。許可の付与の詳細については、「Lake Formation 許可の管理」を参照してください。

5. フェデレーションデータベースのクエリ

   アクセス許可の付与後、ユーザーは Athena および Amazon Redshift を使用してサインインし、フェデレーションデータベースへのクエリを開始できます。これで、ユーザーはローカルデータベース名を使用して SQL クエリで Hive データベースを参照できるようになります。

   Amazon Athena クエリ構文の例

   `fed_glue_db` は、前の手順で作成したローカルデータベース名に置き換えます。

   ```
   Select * from fed_glue_db.customers limit 10;
   ```

その他のリソース

以下のブログ記事には、Hive メタストアデータベースとテーブルに Lake Formation 許可を設定し、Athena を使用してクエリを実行する方法の詳細が記載されています。また、プロデューサーアカウント A の Lake Formation プリンシパルが、LF タグを使用してフェデレーション Hive データベースとテーブルをコンシューマーアカウント B と共有する、クロスアカウント共有のユースケースについても説明します。

- アクセス AWS Lake Formation 許可を使用して Apache Hive メタストアをクエリする

Lake Formation 許可の管理

Lake Formation は、データレイク内のデータに対して一元的なアクセス制御を提供します。Lake Formation ではロールごとにユーザーとアプリケーションのセキュリティポリシーベースのルールを定義でき、 AWS Identity and Access Management との統合によってこれらのユーザーとロールが認証されます。ルールが定義されると、Lake Formation は、Amazon Redshift Spectrum および Amazon Athena のユーザーに対してテーブルレベル、列レベル、行レベルの詳細度でアクセス制御を適用します。

トピック

- データロケーション許可の付与
- データカタログリソースに対するアクセス許可の付与
- 許可のシナリオ例
- Lake Formation でのデータフィルタリングとセルレベルのセキュリティ
- Lake Formation でのデータベースとテーブル許可の表示
- Lake Formation コンソールを使用した許可の取り消し
- Lake Formation でのクロスアカウントデータ共有
- 共有 Data Catalog テーブルとデータベースへのアクセスと表示
- リソースリンクの作成
- クロスリージョンのテーブルアクセス

データロケーション許可の付与

のデータロケーション許可 AWS Lake Formation により、プリンシパルは、指定された登録済み Amazon S3 ロケーションを指す Data Catalog リソースを作成および変更できます。データロケーション許可には、Lake Formation のデータ許可に加えて、データレイク内の情報をセキュア化する働きがあります。

Lake Formation は、データロケーション許可の付与に AWS Resource Access Manager (AWS RAM) サービスを使用しないため、データロケーション許可のリソース共有の招待を受け入れる必要はありません。

データロケーション許可は、Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して付与することができます。

> **Note**
>
> 付与を成功させるには、まずデータロケーションを Lake Formation に登録する必要があります。

> **以下も参照してください。**
>
> - Underlying data access control

トピック

- データロケーション許可の付与 (同じアカウント)
- データロケーション許可の付与 (外部アカウント)
- アカウントと共有されたデータロケーションに対する許可の付与

データロケーション許可の付与 (同じアカウント)

これらの手順を実行して、 AWS アカウント内のプリンシパルにデータロケーション許可を付与します。許可は、Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して付与することができます。

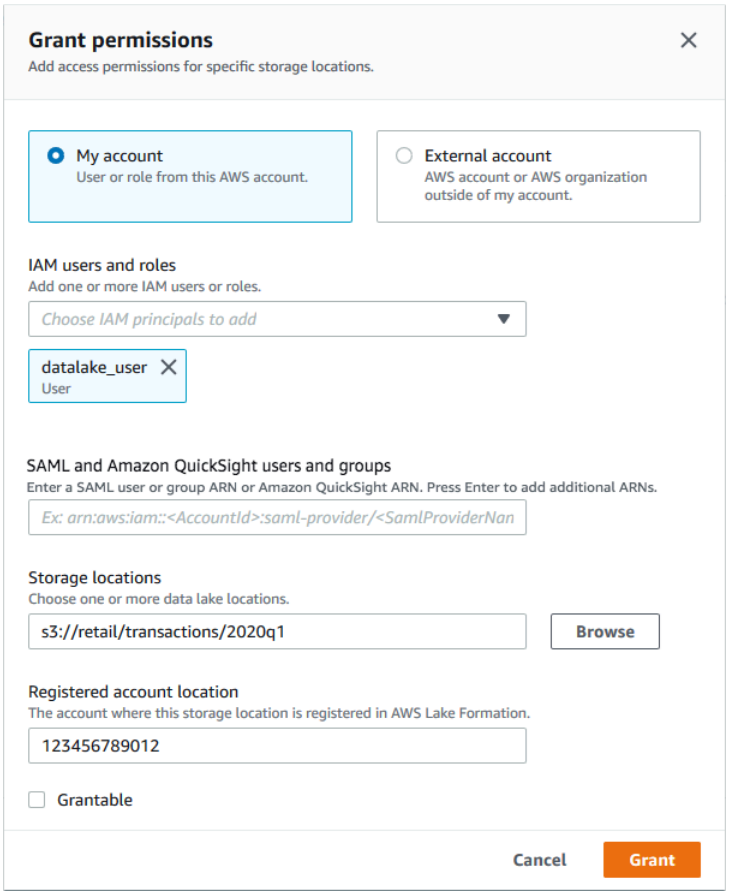
AWS マネジメントコンソール

データロケーションアクセス許可を付与する (同じアカウント)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者、または目的のデータロケーションに対する付与許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインの [Permissions] (許可) で [Data locations] (データのロケーション) を選択します。
3. [Grant] (付与) を選択します。
4. [Grant permissions] (許可の付与) ダイアログボックスで、[My account] (マイアカウント) タイルが選択されていることを確認します。その後、以下の情報を指定します。

- [IAM users and roles] (IAM ユーザーおよびロール) で、1 つ、または複数のプリンシパルを選択します。
- SAML および Amazon Quick ユーザーおよびグループの場合は、SAML または Amazon Quick ユーザーまたはグループの ARNs を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs) を入力します。

  ARN は 1 度に 1 つずつ入力し、各 ARN の後で [Enter] キーを押します。ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。
- [Storage locations] (ストレージのロケーション) では、[Browse] (参照) を選択して、Amazon Simple Storage Service (Amazon S3) ストレージロケーションを選択します。ロケーションは Lake Formation に登録されている必要があります。[Browse] (参照) をもう一度選択して、別のロケーションを追加します。ロケーションは入力することもできますが、ロケーションの前に `s3://` を付けるようにしてください。
- 登録済みアカウントの場所には、その場所が登録されている AWS アカウント ID を入力します。これは、デフォルトでお使いのアカウント ID に設定されます。クロスアカウントのシナリオの場合、受領者アカウントのデータレイク管理者は、受領者アカウント内の他のプリンシパルにデータロケーション許可を付与するときに、ここで所有者アカウントを指定できます。
- (オプション) 選択したプリンシパルが選択したロケーションに対するデータロケーションの許可を付与できるようにするには、[Grantable] (付与可能) を選択します。



5. [Grant] (付与) を選択します。

AWS CLI

データロケーションアクセス許可を付与する (同じアカウント)

- Amazon S3 のパスをリソースとして指定して、`grant-permissions` コマンドを実行し、プリンシパルに `DATA_LOCATION_ACCESS` を付与します。

  Example

  以下の例は、`s3://retail` に対するデータロケーション許可をユーザー `datalake_user1` に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:user/datalake_user1
```

```
 --permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"ResourceArn":"arn:aws:s3:::retail"}}'
```

Example

以下の例は、s3://retail に対するデータロケーションのアクセス許可を ALLIAMPrincipals グループに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333:IAMPrincipals --
permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"ResourceArn":"arn:aws:s3:::retail", "CatalogId": "111122223333"}}'
```

以下も参照してください。

- Lake Formation 許可のリファレンス

データロケーション許可の付与 (外部アカウント)

外部 AWS アカウントまたは組織にデータロケーションのアクセス許可を付与するには、次の手順に従います。

許可は、Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して付与することができます。

[開始する前に]

クロスアカウントアクセスのすべての前提条件が満たされていることを確認します。詳細については、「前提条件」を参照してください。

AWS マネジメントコンソール

データロケーション許可を付与する (外部アカウント、コンソール)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者としてサインインします。
2. ナビゲーションペインの [アクセス許可] で、[データのアクセス許可] を選択し、[付与] を選択します。

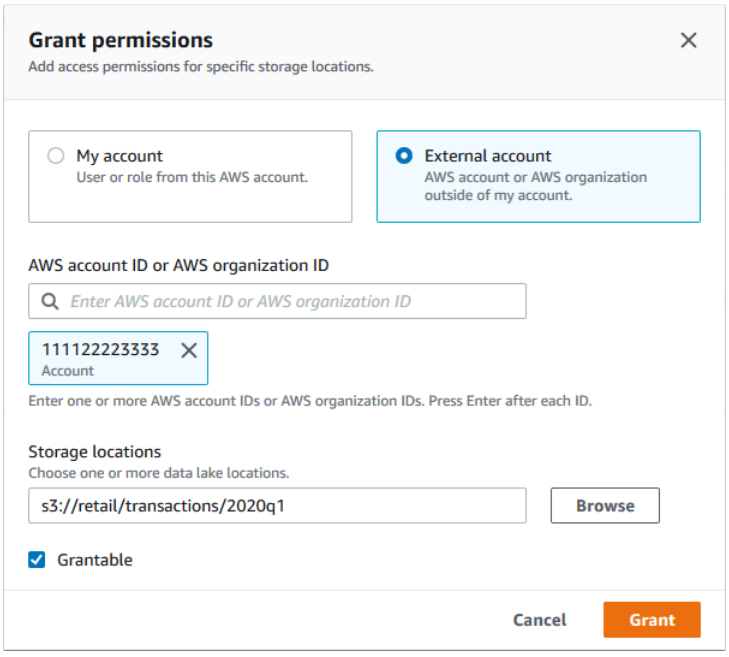
3. [Grant permissions] (許可の付与) ダイアログボックスで、[External account] (外部アカウント) タイルを選択します。
4. 以下の情報を指定します。

   - AWS アカウント ID または AWS 組織 ID には、有効な AWS アカウント番号、組織 IDs、または組織単位 IDsを入力します。

     各 ID の後で [Enter] キーを押します。

     組織 ID は、最初の「o-」と、その後に続く 10～32 個の小文字または数字で構成されています。

     組織単位 ID は、最初の「ou-」と、その後に続く 4～32 個の小文字または数字で構成されています (OU が含まれるルートの ID)。この文字列の後には、2 番目の「-」(ハイフン) と 8～32 個の追加の小文字または数字が続きます。
   - [Storage locations] (ストレージのロケーション) で [Browse] (参照) を選択して、Amazon Simple Storage Service (Amazon S3) ストレージロケーションを選択します。ロケーションは Lake Formation に登録されている必要があります。



5. [Grantable] (付与可能) を選択します。
6. [Grant] (付与) を選択します。

AWS CLI

データロケーションのアクセス許可を付与するには (外部アカウント、 AWS CLI)

- 外部 AWS アカウントにアクセス許可を付与するには、次のようなコマンドを入力します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333  --permissions
 "DATA_LOCATION_ACCESS" --permissions-with-grant-option
 "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"CatalogId":"123456789012","ResourceArn":"arn:aws:s3::retail/
transactions/2020q1"}}'
```

このコマンドは、アカウント 1234-5678-9012 が所有する Amazon S3 ロケーション `s3://retail/transactions/2020q1` に対する grant オプション付きの `DATA_LOCATION_ACCESS` を、アカウント 1111-2222-3333 に付与します。

組織に許可を付与するには、以下のようなコマンドを入力します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:organizations::111122223333:organization/
o-abcdefghijkl --permissions "DATA_LOCATION_ACCESS" --permissions-
with-grant-option "DATA_LOCATION_ACCESS" --resource '{"DataLocation":
 {"CatalogId":"123456789012","ResourceArn":"arn:aws:s3::retail/
transactions/2020q1"}}'
```

このコマンドは、アカウント 1234-5678-9012 が所有する Amazon S3 ロケーション `s3://retail/transactions/2020q1` に対する grant オプション付きの `DATA_LOCATION_ACCESS` を、組織 `o-abcdefghijkl` に付与します。

外部 AWS アカウントのプリンシパルにアクセス許可を付与するには、次のようなコマンドを入力します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"ResourceArn":"arn:aws:s3::retail/transactions/2020q1", "CatalogId":
 "123456789012"}}'
```

このコマンドは、アカウント 1234-5678-9012 が所有する Amazon S3 ロケーション `s3://retail/transactions/2020q1` のアカウント 1111-2222-3333 のプリンシパルに、`DATA_LOCATION_ACCESS` を付与します。

Example

以下の例は、`s3://retail` に対するデータロケーションのアクセス許可を、外部アカウントの `ALLIAMPrincipals` グループに付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333:IAMPrincipals --
permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"ResourceArn":"arn:aws:s3:::retail", "CatalogId": "123456789012"}}'
```

以下も参照してください。

- Lake Formation 許可のリファレンス

アカウントと共有されたデータロケーションに対する許可の付与

Data Catalog リソースが AWS アカウントと共有されると、データレイク管理者として、アカウントの他のプリンシパルにリソースに対するアクセス許可を付与できます。共有テーブルに対する `ALTER` 許可が付与されており、そのテーブルが登録された Amazon S3 ロケーションをポイントする場合は、そのロケーションに対するデータロケーション許可も付与する必要があります。同様に、共有データベースに対する `CREATE_TABLE` または `ALTER` 許可が付与されており、そのデータベースに登録されたロケーションをポイントするロケーションプロパティがある場合は、そのロケーションに対するデータロケーション許可も付与する必要があります。

共有ロケーションに対するデータロケーション許可をアカウント内のプリンシパルに付与するには、そのロケーションに対する grant オプション付きの `DATA_LOCATION_ACCESS` 許可がアカウントに付与されている必要があります。アカウント内の別のプリンシパル`DATA_LOCATION_ACCESS`に を付与するときは、所有者アカウントのデータカタログ ID (AWS アカウント ID) を含める必要があります。所有者アカウントは、ロケーションを登録したアカウントです。

AWS Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して、データロケーションのアクセス許可を付与できます。

アカウントと共有されたデータロケーションに対する許可を付与する (コンソール)

- 「データロケーション許可の付与 (同じアカウント)」の手順を実行します。

  [Storage locations] (ストレージのロケーション) には、ロケーションを入力する必要があります。登録済みアカウントの場所には、所有者 AWS アカウントのアカウント ID を入力します。

アカウントと共有されたデータロケーションに対する許可を付与する (AWS CLI)

- 以下のコマンドのいずれかを入力して、ユーザーまたはロールに許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:user/<user-name>
 --permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"CatalogId":"<owner-account-ID>","ResourceArn":"arn:aws:s3:::<s3-location>"}}'
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:role/<role-name>
 --permissions "DATA_LOCATION_ACCESS" --resource '{ "DataLocation":
 {"CatalogId":"<owner-account-ID>","ResourceArn":"arn:aws:s3:::<s3-location>"}}'
```

データカタログリソースに対するアクセス許可の付与

プリンシパルが Data Catalog リソースを作成および管理し、基盤となるデータにアクセスできるように、 のプリンシパルに Data アクセス許可を付与できます。 AWS Lake Formation カタログ、データベース、テーブル、ビューに対するデータレイクアクセス許可を付与できます。テーブルに対する許可を付与する場合、特定のテーブルの列または行へのアクセスを制限して、より細かな粒度のアクセスコントロールを行うことができます。

個々のカタログ、データベース、テーブル、ビューに対するアクセス許可を付与することも、1 回の付与操作で、カタログまたはデータベース内のすべてのデータベース、テーブル、ビューに対するアクセス許可を付与することもできます。データベース内のすべての IAM プリンシパルに対するアクセス許可を付与すると、データベースに対する DESCRIBE アクセス許可を黙示的に付与することになります。その後は、データベースがコンソールの [Databases] (データベース) ページに表示され、GetDatabases API 操作によって返されます。カタログレベルでも同じ原則が適用されます。カタログ内のデータベースに対するアクセス許可を取得すると、そのカタログに対する DESCRIBE アクセス許可も取得します。

> **Important**
>
> 暗黙的なDESCRIBEアクセス許可は、同じ AWS アカウント内の IAM プリンシパルにアクセス許可を付与する場合にのみ適用されます。クロスアカウントリソースの場合、DESCRIBE アクセス許可を明示的に付与する必要があります。属性ベースのアクセス制御 (ABAC) を使用する場合、DESCRIBE アクセス許可の自動付与は適用されません。属性を使用してデータベース内のすべてのテーブルにアクセス許可を付与する場合、Lake Formation はデータベースに DESCRIBE アクセス許可を暗黙的に付与しません。

許可は、名前付きリソース方式、または Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方式を使用して付与することができます。

同じ のプリンシパル、 AWS アカウント または外部アカウントや組織にアクセス許可を付与できます。外部のアカウントまたは組織に付与するときは、所有するデータカタログオブジェクトをこれらのアカウントまたは組織と共有することになります。このため、これらのアカウントまたは組織のプリンシパルが、所有するデータカタログオブジェクトと、基盤となるデータにアクセスできるようになります。

> **Note**
>
> 現在、LF-TBAC メソッドは、IAM プリンシパル AWS アカウント、組織、組織単位 (OUs) へのクロスアカウントアクセス許可の付与をサポートしています。

外部のアカウントまたは組織に許可を付与する場合は、grant オプションを含める必要があります。共有オブジェクトにアクセスできるのは、外部アカウント内のデータレイク管理者が外部アカウント内の他のプリンシパルに共有オブジェクトに対する許可を付与するまでは、データレイク管理者のみになります。

AWS Lake Formation コンソール、 API、または () を使用して、Data Catalog の AWS Command Line Interface アクセス許可を付与できますAWS CLI。

> **Note**
>
> データカタログオブジェクトを削除すると、そのオブジェクトに関連付けられているすべてのアクセス許可が無効になります。同じリソースを同じ名前で再作成しても、Lake

Formation のアクセス許可は回復しません。ユーザーは新しいアクセス許可を再度設定する必要があります。

以下も参照してください。

- AWS アカウント間でのデータカタログテーブルとデータベースの共有
- メタデータのアクセスコントロール
- Lake Formation 許可のリファレンス

Lake Formation 許可の付与と取り消しに必要な IAM 許可

データレイク管理者を含むすべてのプリンシパルは、Lake Formation API または を使用して AWS Lake Formation Data Catalog アクセス許可またはデータロケーションアクセス許可を付与または取り消すために、次の AWS Identity and Access Management (IAM) アクセス許可が必要です AWS CLI。

- `lakeformation:GrantPermissions`
- `lakeformation:BatchGrantPermissions`
- `lakeformation:RevokePermissions`
- `lakeformation:BatchRevokePermissions`
- 名前付きリソース方式を使用してアクセス許可を付与しているテーブルまたはデータベースの場合は `glue:GetTable`、`glue:GetDatabase`、または`glue:GetCatalog`。

Note

データレイク管理者には Lake Formation 許可を付与して取り消すための黙示的な Lake Formation 許可がありますが、それでも Lake Formation の付与および取り消し API 操作に対する IAM 許可が必要です。
`AWSLakeFormationDataAdmin` AWS マネージドポリシーを持つ IAM ロールは、Lake Formation API オペレーション に対する明示的な拒否が含まれているため、新しいデータレイク管理者を追加できません`PutDataLakeSetting`。

以下の IAM ポリシーは、データレイク管理者ではないが、Lake Formation コンソールを使用して許可を付与または取り消したいというプリンシパルに推奨されます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:ListPermissions",
                "lakeformation:GrantPermissions",
                "lakeformation:BatchGrantPermissions",
                "lakeformation:RevokePermissions",
                "lakeformation:BatchRevokePermissions",
                "glue:GetCatalogs",
                "glue:GetDatabases",
                "glue:SearchTables",
                "glue:GetTables",
                "glue:GetCatalog",
                "glue:GetDatabase",
                "glue:GetTable",
                "iam:ListUsers",
                "iam:ListRoles",
                "sso-directory:DescribeUser",
                "sso-directory:DescribeGroup",
                "sso:DescribeInstance"
            ],
            "Resource": "*"
        }
    ]
}
```

このポリシーのすべての `glue:`および アクセス`iam:`許可は、 AWS 管理ポリシー で使用できます`AWSGlueConsoleFullAccess`。

Lake Formation のタグベースのアクセス制御 (LF-TBAC) を使用して許可を付与するには、プリンシパルに追加の IAM 許可が必要です。詳細については、「Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項」および「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

クロスアカウントアクセス許可

名前付きリソースメソッドを使用してクロスアカウント Lake Formation アクセス許可を付与するユーザーには、 `AWSLakeFormationCrossAccountManager` AWS マネージドポリシーの アクセス許可も必要です。

データレイク管理者には、クロスアカウントアクセス許可を付与するための同じアクセス許可と、組織へのアクセス許可の付与を有効にするための AWS Resource Access Manager (AWS RAM) アクセス許可が必要です。詳細については、「データレイク管理者の許可」を参照してください。

管理ユーザー

`AdministratorAccess` AWS 管理ポリシーなどの管理アクセス許可を持つプリンシパルには、Lake Formation アクセス許可を付与し、データレイク管理者を作成するアクセス許可があります。Lake Formation 管理者操作へのユーザーまたはロールのアクセスを拒否するには、そのポリシーに管理者 API 操作の `Deny` ステートメントをアタッチまたは追加してください。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "lakeformation:GetDataLakeSettings",
                "lakeformation:PutDataLakeSettings"
            ],
            "Effect": "Deny",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Important

ユーザーが抽出、変換、ロード (ETL) スクリプトを使用してユーザー自身を管理者として追加できないようにするには、管理者以外のすべてのユーザーとロールに対してこれらの API 操作へのアクセスが拒否されていることを確認してくださ

い。AWSLakeFormationDataAdmin AWS 管理ポリシーには、ユーザーが新しいデータレイク管理者を追加できないPutDataLakeSettingように、Lake Formation API オペレーションの明示的な拒否が含まれています。

名前付きリソース方式を使用したデータアクセス許可の付与

名前付きデータカタログリソース方式は、一元化されたアプローチを使用して、データベース、テーブル、列、ビューなどの AWS Glue Data Catalog オブジェクトにアクセス許可を付与する方法です。これにより、データレイク内の特定のリソースへのアクセスを制御するリソースベースのポリシーを定義できます。

名前付きリソース方式を使用してアクセス許可を付与するときには、リソースタイプと、そのリソースに対して付与または取り消すアクセス許可を指定できます。必要に応じて後からアクセス許可を取り消して、関連付けられているリソースからアクセス許可を削除することもできます。

AWS Lake Formation コンソール、APIs、または AWS Command Line Interface () を使用してアクセス許可を付与できますAWS CLI。

トピック

- 名前付きリソース方式を使用したデータカタログアクセス許可の付与
- 名前付きリソース方式を使用したデータベースのアクセス許可の付与
- 名前付きリソース方式を使用したテーブル許可の付与
- 名前付きリソース方式を使用したビューに対するアクセス権限の付与

名前付きリソース方式を使用したデータカタログアクセス許可の付与

以下は、名前付きリソース方式を使用してカタログアクセス許可を付与する方法を説明する手順です。

Console

Lake Formation コンソールの [アクセス許可を付与] ページを使用します。このページは、以下のセクションに分かれています。

- プリンシパルタイプ – 特定のプリンシパルにアクセス許可を付与することも、属性タグを使用することもできます。

- [プリンシパル] – アクセス許可の付与先となる IAM ユーザー、ロール、IAM アイデンティティセンターユーザーとグループ、 AWS アカウント、組織、または組織単位。

  [属性別のプリンシパル] – IAM ロールまたは IAM セッションタグからタグのキーと値のペアを追加します。一致する属性を持つプリンシパルは、指定されたリソースへのアクセスを受け取ります。
- [LF タグまたはカタログリソース] – 付与するアクセス許可の対象となるカタログ、データベース、テーブル、ビュー、またはリソースリンク。
- [Permissions] (許可) – 付与される Lake Formation 許可。

> **Note**
>
> データベースリソースリンクに対する許可を付与するには、「リソースリンク許可の付与」を参照してください。

1. 「アクセス許可の付与」ページを開きます。

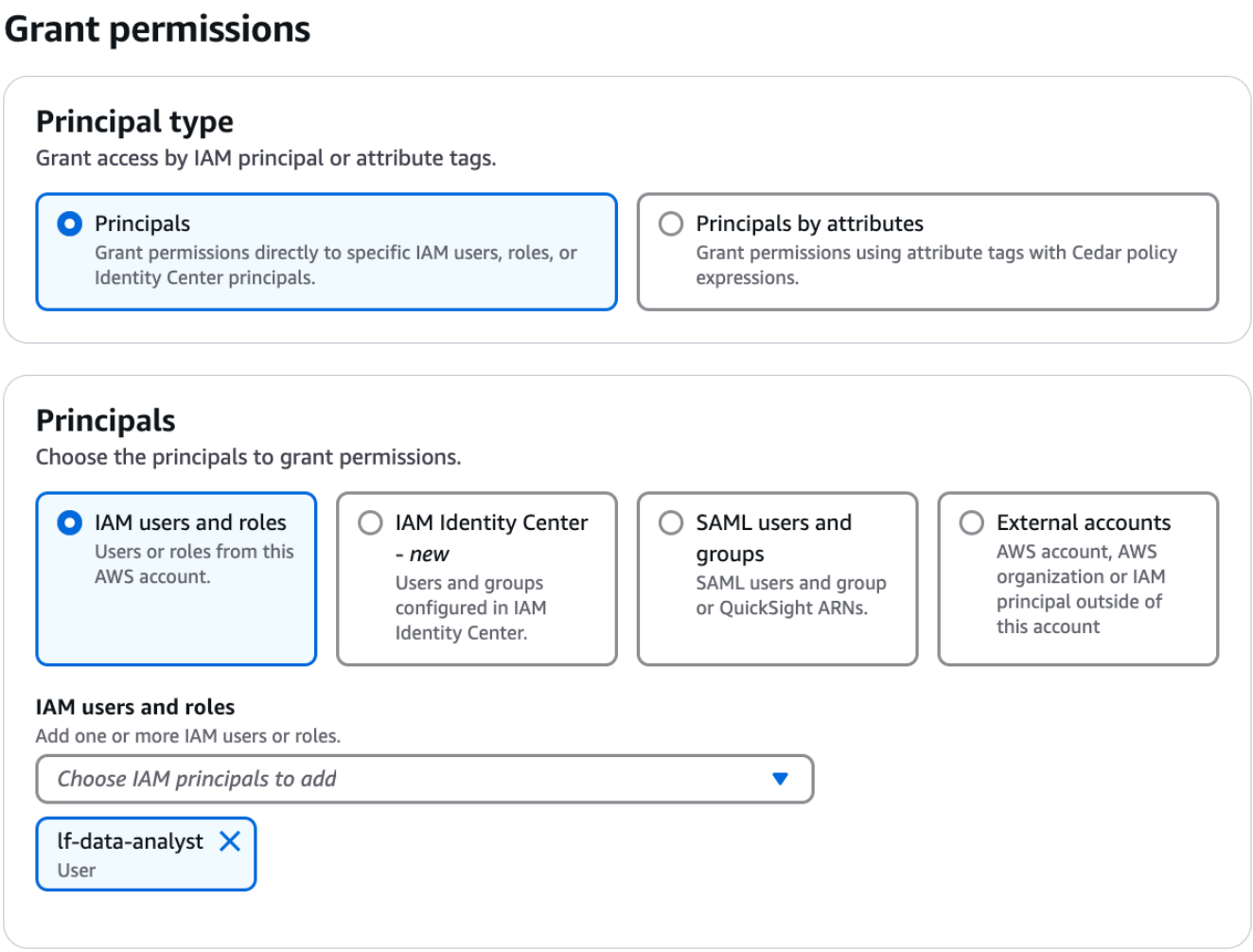
   https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、データレイク管理者、カタログ作成者、またはカタログに対する付与可能なアクセス許可を持つ IAM ユーザーとしてサインインします。

   次のいずれかを行います。

   - ナビゲーションペインの [Permissions] (許可) で [Data permissions] (データの許可) を選択します。次に、[Grant] (付与) を選択します。
   - ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。次に、「カタログ」ページでカタログを選択し、[アクション] メニューの [アクセル許可] で [付与] を選択します。

   > **Note**
   >
   > テーブルに対するアクセス許可は、リソースリンクを使用して付与できます。これを実行するには、[テーブル] ページでカタログリンクコンテナを選択し、[アクション] メニューで [ターゲットに対して付与] を選択します。詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。

2. 次に、[プリンシパルタイプ] セクションで、プリンシパルを選択するか、プリンシパルにアタッチされた属性を指定します。



プリンシパルを指定する

IAM ユーザーとロール

[IAM users and roles] (IAM ユーザーおよびロール) リストから、1 人、または複数のユーザーまたはロールを選択します。

IAM アイデンティティセンター

[ユーザーとグループ] リストから、1 人、または複数のユーザーまたはグループを選択します。ユーザーまたはグループをさらに追加するには、[追加] を選択します。

SAML ユーザーとグループ

SAML および Quick ユーザーおよびグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、または Amazon Quick ユーザーまたはグループの ARNs を入力します。各 ARN の後で Enter キーを押します。

ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。

> **Note**
>
> Lake Formation と Quick の統合は、Quick Enterprise Edition でのみサポートされています。

外部アカウント

AWS アカウント、 AWS 組織、または IAM プリンシパルには、IAM ユーザーまたはロールの 1 つ以上の有効な AWS アカウント IDs、組織 IDs、組織単位 IDs、または ARN を入力します。各 ID の後で [Enter] キーを押します。

組織 ID は、最初の「o-」と、その後に続く 10～32 個の小文字または数字で構成されています。

組織単位 ID は「ou-」で始まり、その後に 4～32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」ダッシュと 8～32 個の追加の小文字または数字が続きます。

属性別のプリンシパル

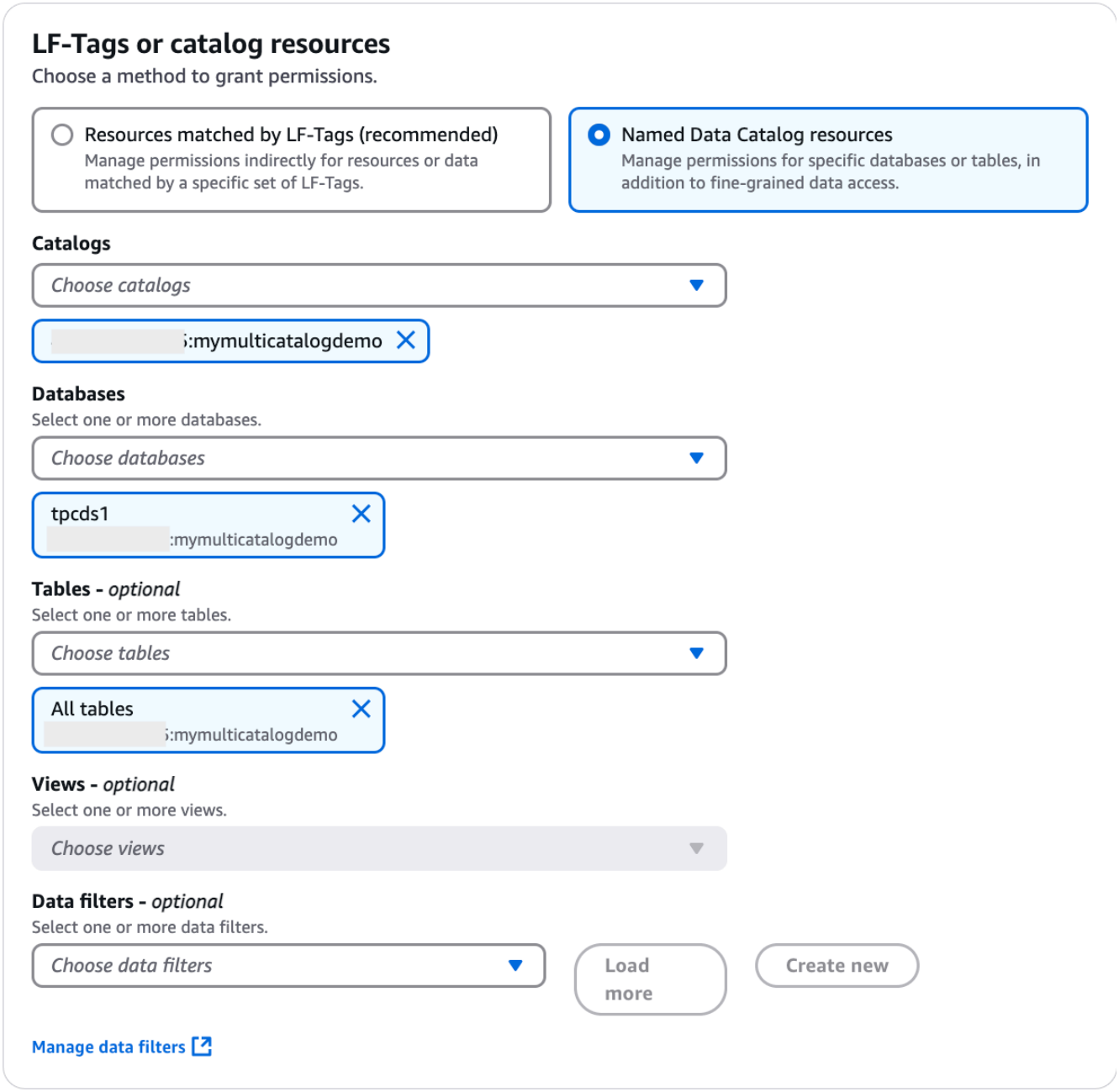
属性

IAM ロールから IAM タグのキーと値のペアを追加します。

アクセス許可のスコープ

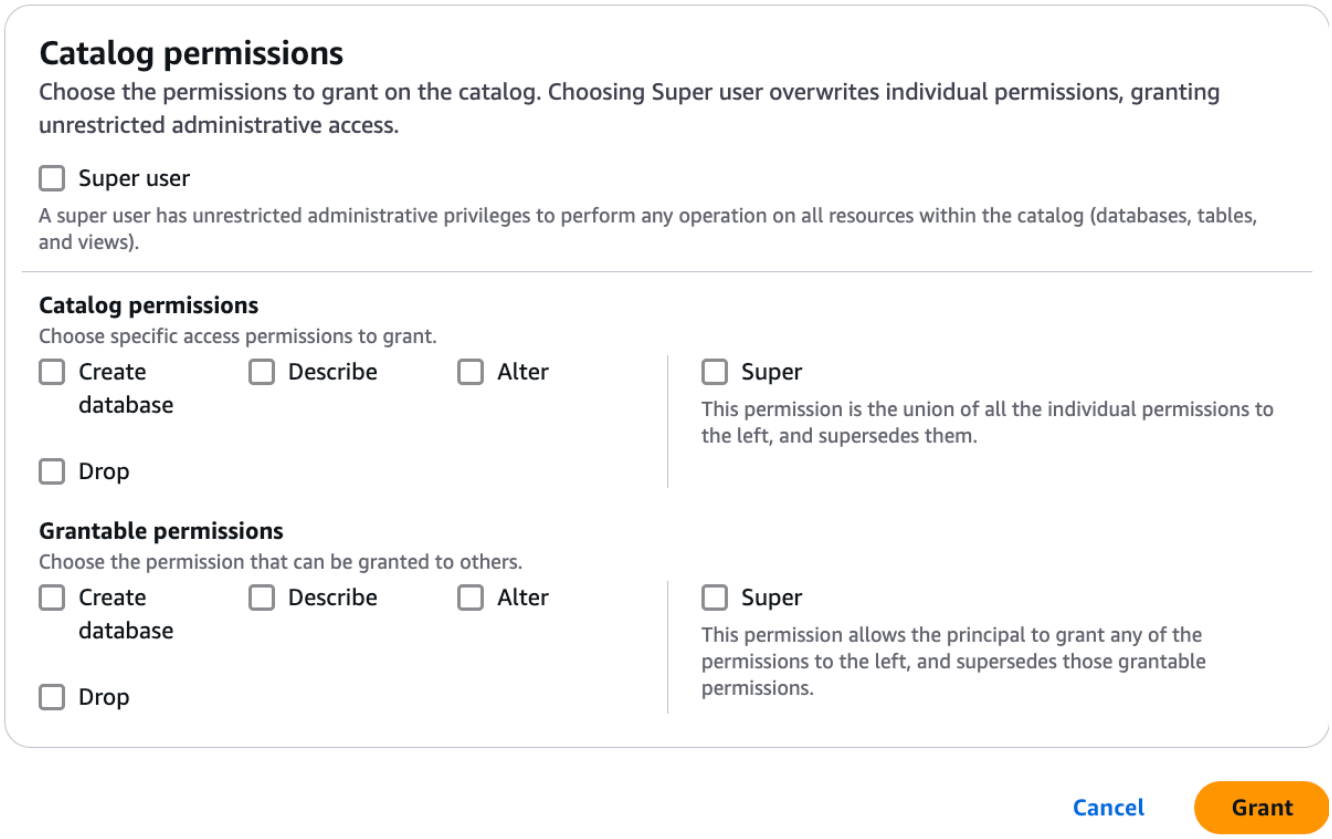
同じアカウントまたは別のアカウントの一致する属性を持つプリンシパルにアクセス許可を付与するかどうかを指定します。

3. [LF タグまたはカタログリソース] セクションで、[名前付きのデータカタログリソース] を選択します。



4. [カタログ] リストから 1 つ以上のカタログを選択します。1 つ以上のデータベース、テーブル、または データフィルターを選択することもできます。

5. [アクセス許可] セクションで、アクセス許可と付与可能なアクセス許可を選択します。[カタログのアクセス許可] で、付与するアクセス許可を 1 つ以上選択します。



[スーパーユーザー] を選択して、カタログ内のすべてのリソース (データベース、テーブル、ビュー) に対して任意のオペレーションを実行するための無制限の管理権限を付与します。

> **Note**
>
> 登録済みの場所を指す場所プロパティを持つカタログに対して `Create database` または `Alter` を付与した後は、プリンシパルにもその場所に対するデータの場所のアクセス許可を付与するようにしてください。詳細については、「データロケーション許可の付与」を参照してください。

6. (オプション) [Grantable permissions] (付与可能な許可) で、付与対象者がそれぞれの AWS アカウント内の他のプリンシパルに付与できる許可を選択します。このオプションは、外部アカウントから IAM プリンシパルにアクセス許可を付与する場合はサポートされません。
7. [付与] を選択します。

「データアクセスの許可」ページには、アクセス許可の詳細が表示されます。[属性別のプリンシパル] オプションを使用してアクセス許可を付与した場合、`ALLPrincipals` に対するアクセス許可の付与をリストに表示できます。

AWS CLI

を使用したカタログアクセス許可の付与については AWS CLI、「」を参照してくださいAmazon Redshift フェデレーティッドカタログの作成。

名前付きリソース方式を使用したデータベースのアクセス許可の付与

以下は、名前付きリソース方式を使用してデータベース許可を付与する方法を説明する手順です。

Console

Lake Formation コンソールの「アクセス許可を付与」ページを使用します。このページは、以下のセクションに分かれています。

- プリンシパルタイプ – [プリンシパル] セクションには、アクセス許可の付与先となる IAM ユーザー、ロール、IAM アイデンティティセンターのユーザーとグループ、 AWS アカウント、組織、または組織単位が含まれます。[属性別のプリンシパル] セクションでは、IAM ロールにアタッチされた属性のキーと値を指定できます。
- [LF タグまたはカタログリソース] – 付与する許可の対象となるデータベース、テーブル、ビュー、またはリソースリンク。
- [Permissions] (許可) – 付与される Lake Formation 許可。

> **Note**
>
> データベースリソースリンクに対する許可を付与するには、「リソースリンク許可の付与」を参照してください。

1. 「アクセス許可の付与」ページを開きます。

   https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、データレイク管理者、データベース作成者、またはデータベースに対する付与可能なアクセス許可を持つ IAM ユーザーとしてサインインします。

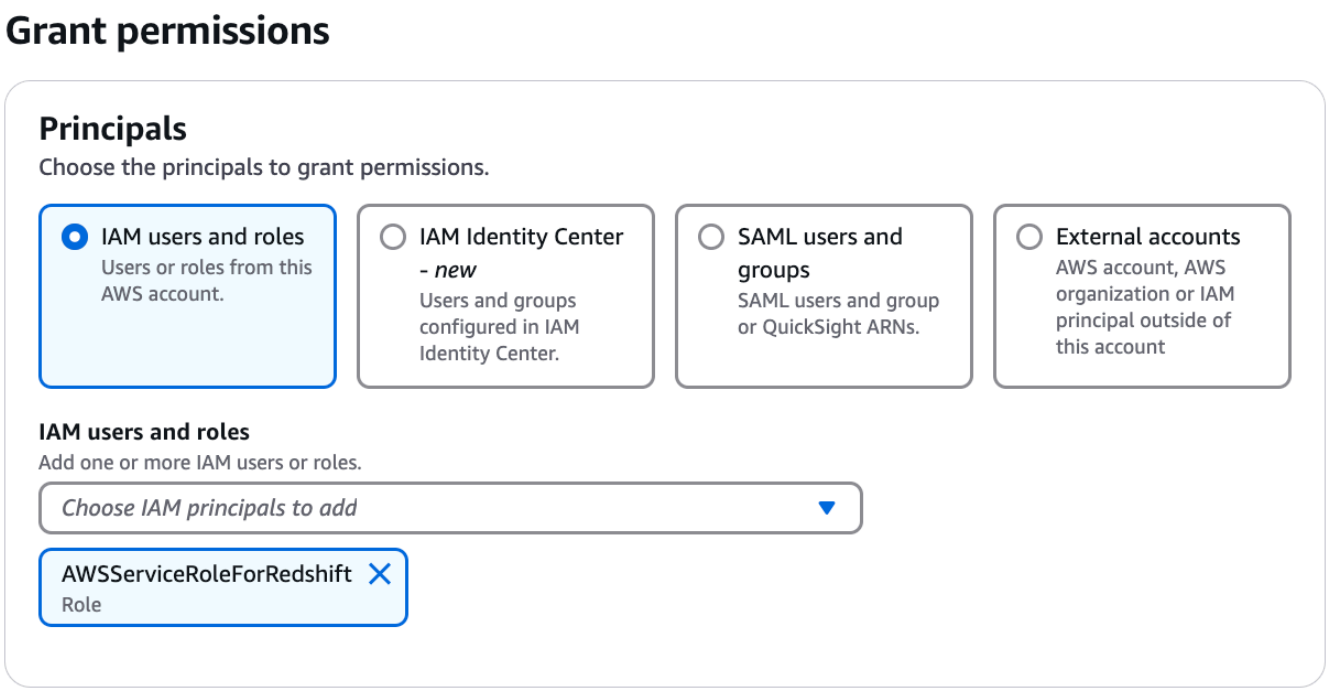
   次のいずれかを行います。

   - ナビゲーションペインの [Permissions] (許可) で [Data permissions] (データの許可) を選択します。次に、[Grant] (付与) を選択します。

- ナビゲーションペインの [データカタログ] で [データベース] を選択します。次に、[データベース] ページでデータベースを選択し、[アクション] メニューの [許可] で [付与] を選択します。

> **Note**
>
> データベースに対する許可は、そのリソースリンクを使用して付与できます。これを実行するには、[Database] (データベース) ページでリソースリンクを選択し、[Actions] (アクション) メニューで [Grant on target] (ターゲットに対して付与) を選択します。詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。

2. [プリンシパル] セクションで、プリンシパルを指定するか、属性を使用してプリンシパルにアクセス許可を付与します。



IAM ユーザーとロール

[IAM users and roles] (IAM ユーザーおよびロール) リストから、1 人、または複数のユーザーまたはロールを選択します。

IAM アイデンティティセンター

[ユーザーとグループ] リストから、1 人、または複数のユーザーまたはグループを選択します。ユーザーまたはグループをさらに追加するには、[追加] を選択します。

SAML ユーザーとグループ

SAML および Quick ユーザーおよびグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、または Amazon Quick ユーザーまたはグループの ARNs を入力します。各 ARN の後で Enter キーを押します。

ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。

> **Note**
>
> Lake Formation と Quick の統合は、Quick Enterprise Edition でのみサポートされています。

外部アカウント

AWS アカウント、 AWS 組織、または IAM プリンシパルには、IAM ユーザーまたはロールの 1 つ以上の有効な AWS アカウント IDs、組織 IDs、組織単位 IDs、または ARN を入力します。各 ID の後で [Enter] キーを押します。

組織 ID は、最初の「o-」と、その後に続く 10 ~ 32 個の小文字または数字で構成されています。

組織単位 ID は「ou-」で始まり、その後に 4 ~ 32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」ダッシュと 8 ~ 32 個の追加の小文字または数字が続きます。

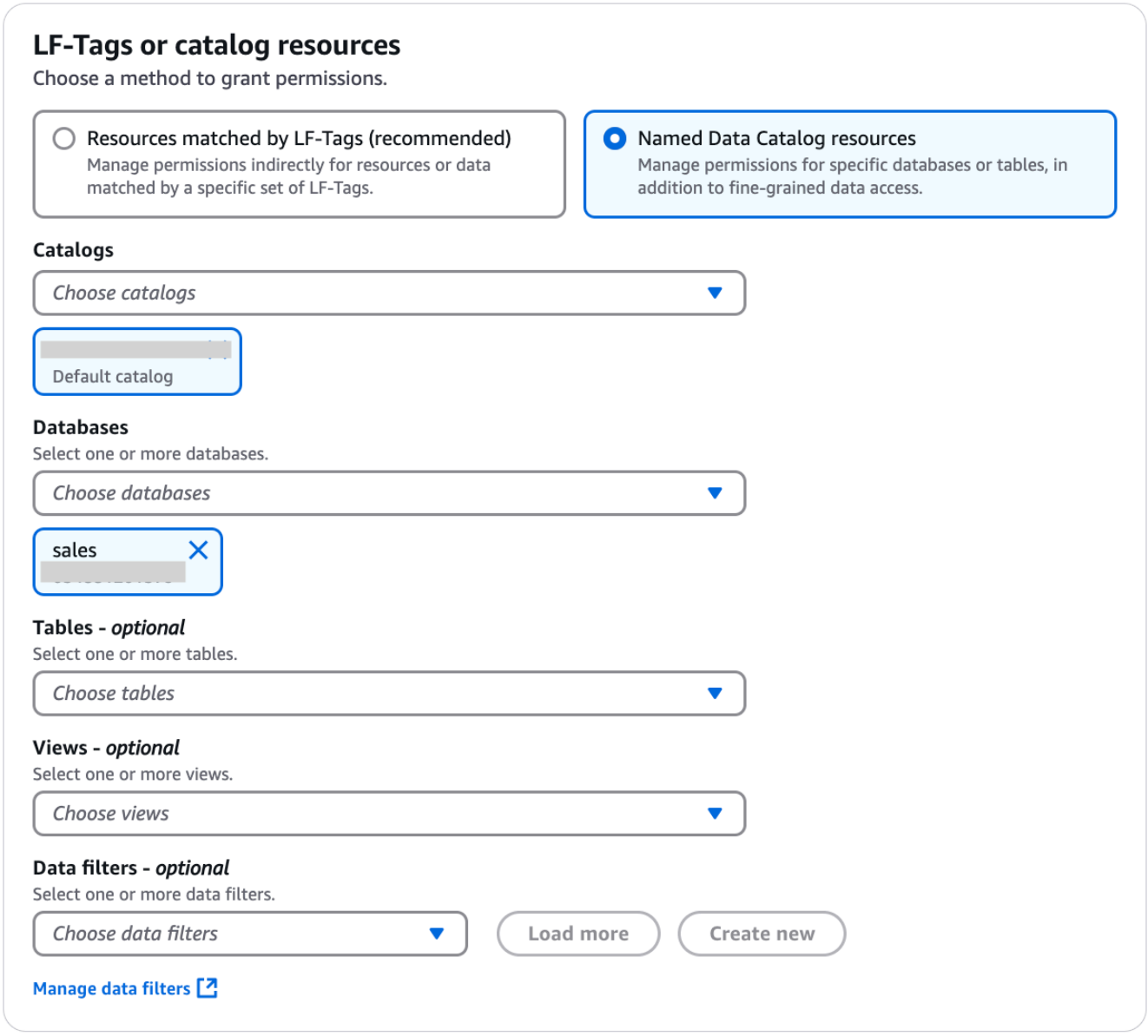
属性別のプリンシパル

属性のキーと値を指定します。複数の値を選択する場合は、OR 演算子で属性式を作成することになります。つまり、IAM ロールまたはユーザーに割り当てられた属性タグ値の

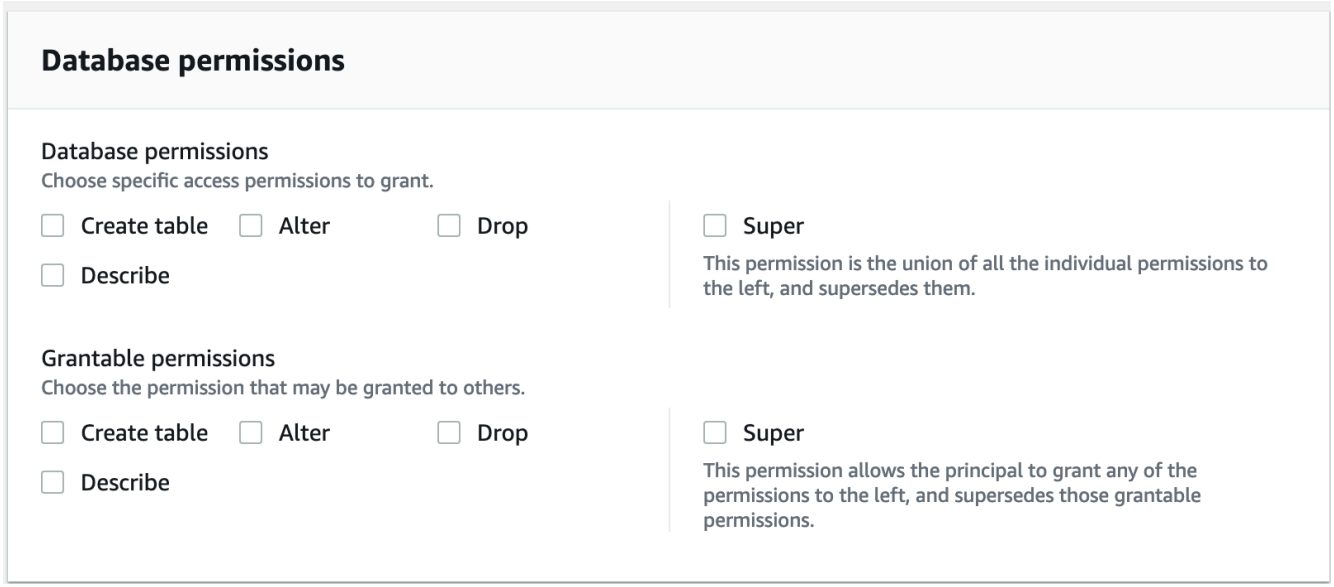
いずれかが一致した場合、ロール/ユーザーはリソースに対するアクセス許可を取得します。

同じアカウントまたは別のアカウントで一致する属性を持つプリンシパルにアクセス許可を付与するかどうかを指定して、アクセス許可の範囲を選択します。

3. [LF タグまたはカタログリソース] セクションで、[名前付きのデータカタログリソース] を選択します。



4. [Database] (データベース) のリストから、1 つ、または複数のデータベースを選択します。1 つ以上のテーブルやデータフィルターを選択することもできます。
5. [Permissions] (許可) セクションで、許可と付与可能な許可を選択します。[Database permissions] (データベースの許可) で、付与する許可を 1 つ、または複数選択します。



> **Note**
>
> 登録されたロケーションをポイントするロケーションプロパティを持ったデータベースに対する `Create Table` または `Alter` を付与した後は、プリンシパルにもそのロケーションに対するデータロケーション許可を付与するようにしてください。詳細については、「データロケーション許可の付与」を参照してください。

6. (オプション) [Grantable permissions] (付与可能な許可) で、付与対象者がそれぞれの AWS アカウント内の他のプリンシパルに付与できる許可を選択します。このオプションは、外部アカウントから IAM プリンシパルにアクセス許可を付与する場合はサポートされません。
7. [付与] を選択します。

AWS CLI

データベース許可は、名前付きリソース方式と AWS Command Line Interface (AWS CLI) を使用して付与することができます。

を使用してデータベースのアクセス許可を付与するには AWS CLI

- `grant-permissions` コマンドを実行し、付与される許可に応じて、データベースまたは Data Catalog をリソースとして指定します。

  次の例では、*`<account-id>`* を有効な AWS アカウント ID に置き換えます。

Example– データベースを作成するための付与

この例では、CREATE_DATABASE をユーザー datalake_user1 に付与します。この許可が付与されるリソースは Data Catalog であるため、コマンドは resource パラメータとして空の CatalogResource 構造を指定します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:user/datalake_user1 --
permissions "CREATE_DATABASE" --resource '{ "Catalog": {}}'
```

Example– 指定されたデータベースでテーブルを作成するための付与

次の例は、データベース retail での CREATE_TABLE をユーザー datalake_user1 に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::<account-id>:user/datalake_user1 --
permissions "CREATE_TABLE" --resource '{ "Database": {"Name":"retail"}}'
```

Example– Grant オプションを使用して外部 AWS アカウントに付与する

次の例は、データベース retail に対する grant オプション付きの CREATE_TABLE を外部アカウント 1111-2222-3333 に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333 --permissions "CREATE_TABLE"
 --permissions-with-grant-option "CREATE_TABLE" --resource '{ "Database":
 {"Name":"retail"}}'
```

Example– 組織への付与

次の例は、データベース issues に対する grant オプション付きの ALTER を組織 o-abcdefghijkl に付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:organizations::111122223333:organization/
o-abcdefghijkl --permissions "ALTER" --permissions-with-grant-option "ALTER" --
resource '{ "Database": {"Name":"issues"}}'
```

Example- 同じアカウントで ALLIAMPrincipals に付与

次の例では、同じアカウントのすべてのプリンシパルにデータベース retail への CREATE_TABLE アクセス許可を付与します。このオプションを使用すると、アカウント内のすべてのプリンシパルがデータベースにテーブルを作成し、統合クエリエンジンが共有データベースとテーブルにアクセスできるようにするテーブルリソースリンクを作成できます。このオプションは、プリンシパルがクロスアカウント付与を受け取っていて、リソースリンクを作成するアクセス許可を持っていない場合に特に役立ちます。このシナリオでは、データレイク管理者がプレースホルダーデータベースを作成して ALLIAMPrincipal グループに CREATE_TABLE アクセス許可を付与し、アカウント内の各 IAM プリンシパルがプレースホルダーデータベースにリソースリンクを作成できるようにします。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333:IAMPrincipals
 --permissions "CREATE_TABLE"  --resource '{ "Database":
 {"Name":"temp","CatalogId":"111122223333"}}'
```

Example- 外部アカウントでの ALLIAMPrincipals への付与

次の例では、外部アカウントのすべてのプリンシパルにデータベース retail への CREATE_TABLE を付与します。このオプションにより、アカウント内の各プリンシパルがデータベースにテーブルを作成できます。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333:IAMPrincipals
 --permissions "CREATE_TABLE"  --resource '{ "Database":
 {"Name":"retail","CatalogId":"123456789012"}}'
```

> **Note**
>
> 登録されたロケーションをポイントするロケーションプロパティを持ったデータベースに対する CREATE_TABLE または ALTER を付与した後は、プリンシパルにもそのロケーションに対するデータロケーション許可を付与するようにしてください。詳細については、「データロケーション許可の付与」を参照してください。

関連情報

- Lake Formation 許可のリファレンス
- アカウントと共有されたデータベースまたはテーブルに対する許可の付与
- 共有 Data Catalog テーブルとデータベースへのアクセスと表示

名前付きリソース方式を使用したテーブル許可の付与

Lake Formation コンソールまたは を使用して AWS CLI 、データカタログテーブルに対する Lake Formation アクセス許可を付与できます。個々のテーブルに対する許可を付与する、または 1 回の付与操作で、データベース内のすべてのテーブルに対する許可を付与することができます。

データベース内のすべてのテーブルに対する許可を付与すると、データベースに対する DESCRIBE 許可を黙示的に付与することになります。その後は、データベースがコンソールの [Databases] (データベース) ページに表示され、GetDatabases API 操作によって返されます。この DESCRIBE アクセス許可の自動付与は、属性ベースのアクセス制御 (ABAC) を使用する場合には適用されません。属性を使用してデータベース内のすべてのテーブルにアクセス許可を付与する場合、Lake Formation はデータベースに DESCRIBE アクセス許可を暗黙的に付与しません。

付与する許可として SELECT を選択するときは、列フィルター、行フィルター、またはセルフィルターを適用するオプションがあります。

Console

以下は、名前付きリソース方式と、Lake Formation コンソールの [データレイクのアクセス許可を付与] ページを使用して、テーブル許可を付与する方法を説明する手順です。このページは、これらのセクションに分けられています。

- プリンシパルタイプ – アクセス許可の付与先となるユーザー、ロール、 AWS アカウント、組織、または組織単位。一致する属性を持つプリンシパルにアクセス許可を付与することもできます。
- [LF-Tags or catalog resources] (LF タグまたはカタログリソース) – 付与する許可の対象となるデータベース、テーブル、またはリソースリンク。
- [Permissions] (許可) – 付与される Lake Formation 許可。

> **Note**
>
> テーブルリソースリンクに対する許可を付与するには、「リソースリンク許可の付与」を参照してください。

1. 「アクセス許可の付与」ページを開きます。

   https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、データレイク管理者、テーブル作成者、またはテーブルに対するアクセス許可を付与されたユーザーとして、グラントオプションを使用してサインインします。

   次のいずれかを行います。

   - ナビゲーションペインの [アクセス許可] で [データの許可] を選択します。次に、[Grant] (付与) を選択します。
   - ナビゲーションペインで [Table] (テーブル) を選択します。次に、[Tables] (テーブル) ページでテーブルを選択し、[Actions] (アクション) メニューの [Permissions] (許可) で [Grant] (付与) を選択します。

   > **Note**
   >
   > テーブルに対する許可は、リソースリンクを使用して付与することができます。これを実行するには、[Tables] (テーブル) ページでリソースリンクを選択し、[Actions] (アクション) メニューで [Grant on target] (ターゲットに対して付与) を選択します。詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。

2. 次に、[プリンシパル] セクションで、プリンシパルまたは一致する属性を持つプリンシパルを指定して、アクセス許可を付与します。

   IAM ユーザーとロール

   [IAM users and roles] (IAM ユーザーおよびロール) リストから、1 人、または複数のユーザーまたはロールを選択します。

IAM アイデンティティセンター

[ユーザーとグループ] リストから、1 人、または複数のユーザーまたはグループを選択します。

SAML ユーザーとグループ

SAML およびクイックユーザーおよびグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、またはクイックユーザーまたはグループの ARNs を入力します。各 ARN の後で Enter キーを押します。

ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。

> **Note**
>
> Lake Formation と Quick の統合は、Quick Enterprise Edition でのみサポートされています。

外部アカウント

AWS アカウント 、 AWS 組織、または IAM プリンシパルには、IAM ユーザーまたはロールの 1 つ以上の AWS アカウント IDs、組織 IDs、組織単位 IDs、または ARN を入力します。各 ID の後で [Enter] キーを押します。

組織 ID は、最初の「o-」と、その後に続く 10 ~ 32 個の小文字または数字で構成されています。

組織単位 ID は「ou-」で始まり、その後に 4 ~ 32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」文字と 8 ~ 32 個の追加の小文字または数字が続きます。

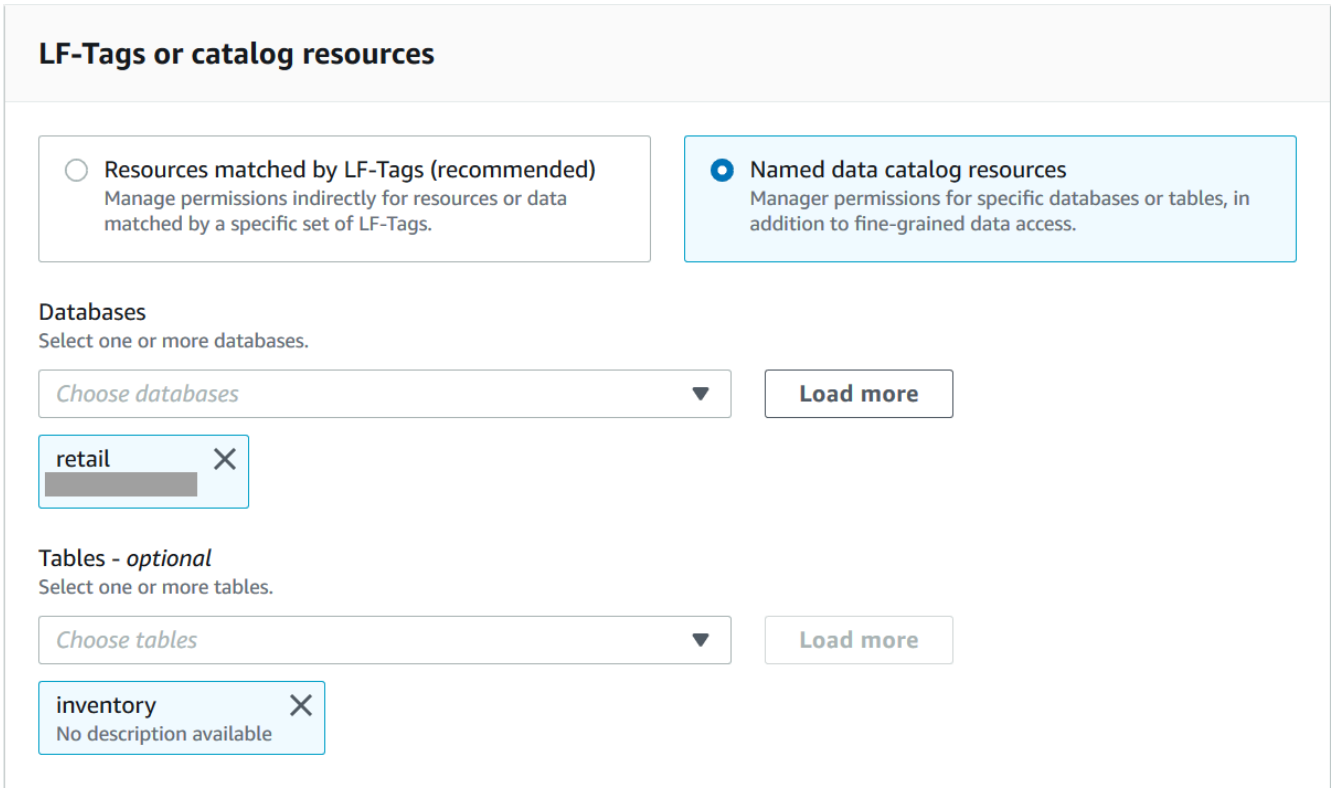
属性別のプリンシパル

属性のキーと値を指定します。複数の値を選択する場合は、OR 演算子で属性式を作成することになります。つまり、IAM ロールまたはユーザーに割り当てられた属性タグ値の

いずれかが一致した場合、ロール/ユーザーはリソースに対するアクセス許可を取得します。

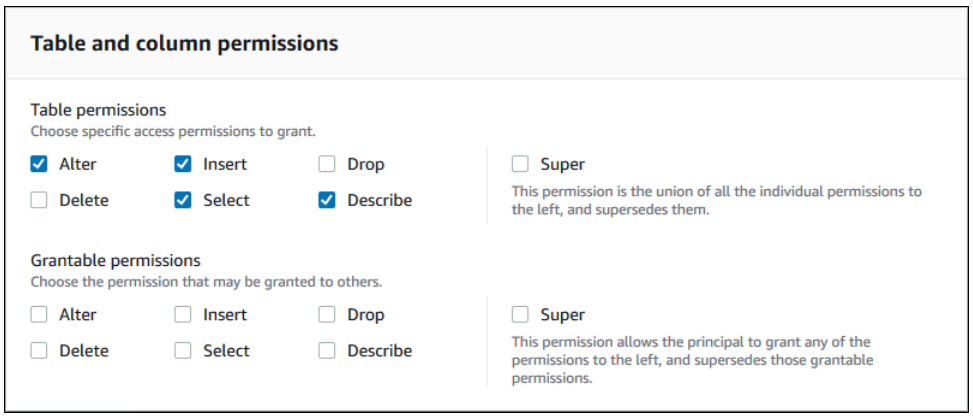
同じアカウントまたは別のアカウントで一致する属性を持つプリンシパルにアクセス許可を付与するかどうかを指定して、アクセス許可の範囲を選択します。

3. [LF-Tags or catalog resources] (LF タグまたはカタログリソース) セクションで、データベースを選択します。次に、1 つ、または複数のテーブルを選択するか、[All tables] (すべてのテーブル) を選択します。

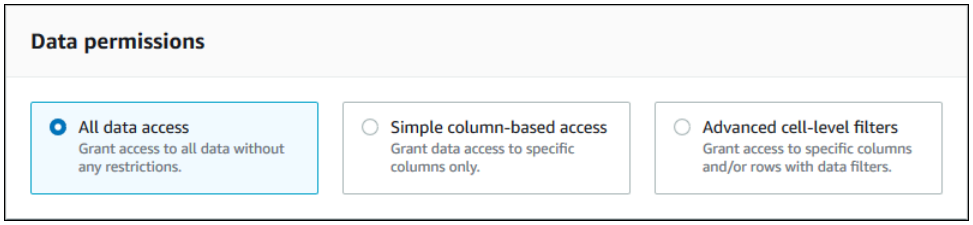


4. データフィルタリングを使用せずにアクセス許可を指定します。

[許可] セクションで、付与するテーブル許可を選択し、オプションで付与可能な許可を選択します。



[選択] を付与する場合は、[テーブルと列のアクセス許可] セクションの下に、[すべてのデータアクセス] オプションがデフォルトで選択された [データの許可] セクションが表示されます。デフォルトを受け入れます。



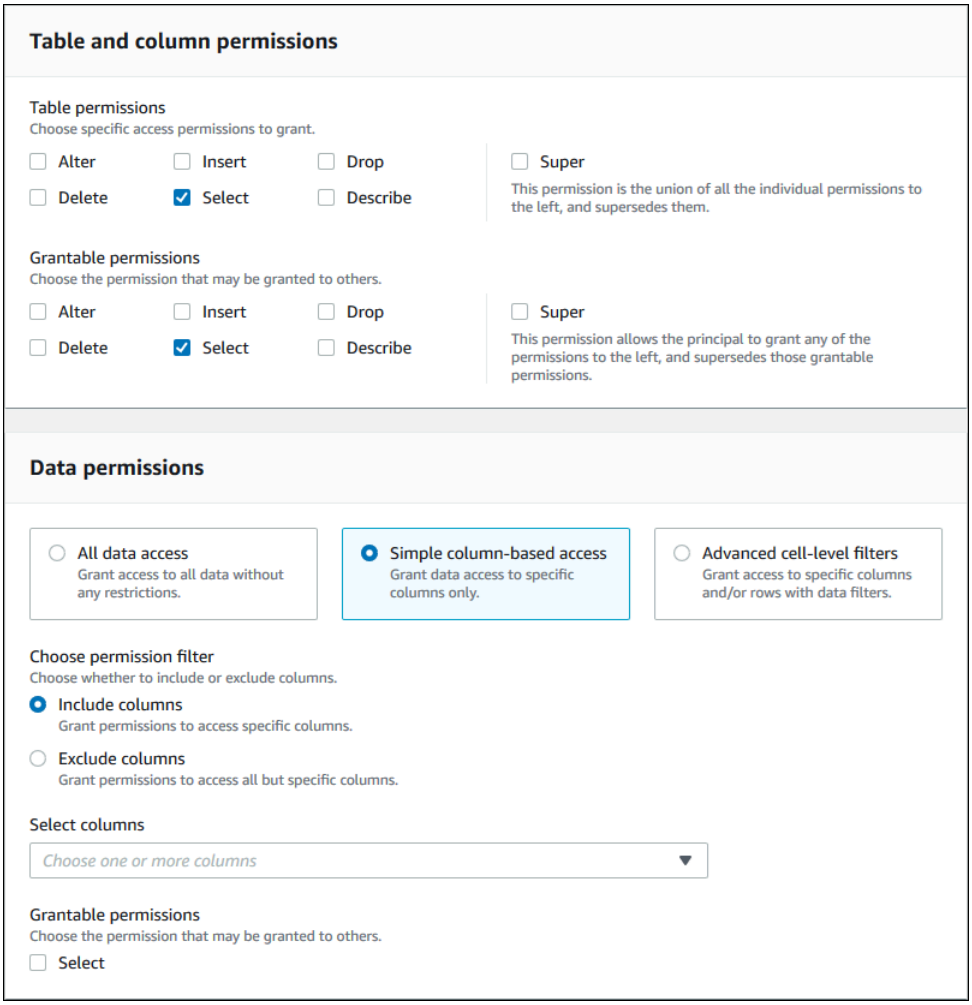
5. [Grant] (付与) を選択します。
6. データフィルタリングを使用して選択許可を指定する

   [Select] (選択) 許可を選択します。他の許可は選択しないでください。

   [Data permissions] (データの許可) セクションが、[Table and column permissions] (テーブルと列の許可) セクションの下に表示されます。
7. 以下のいずれかを実行します。

   - シンプルな列フィルタリングのみを適用します。

     1. [Simple column-based access] (シンプルな列ベースのアクセス) を選択します。



2. 列を含めるか除外するかを選択してから、含める、または除外する列を選択します。

   外部 AWS アカウントまたは組織にアクセス許可を付与する場合、インクルードリストのみがサポートされます。

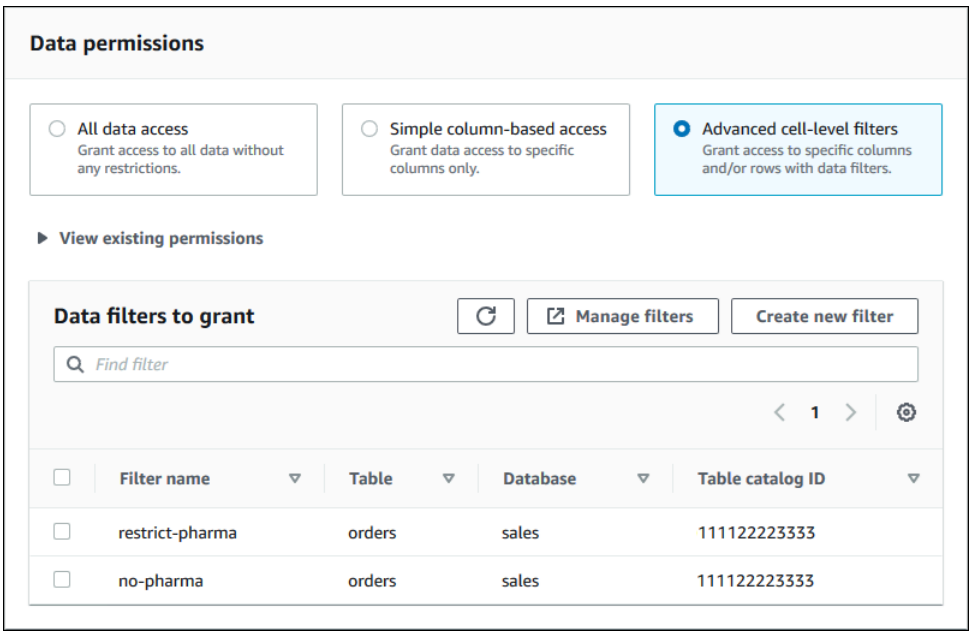
3. (オプション) [Grantable permissions] (付与可能な許可) で、[Select] (選択) に対して grant オプションをオンにします。

   Grant オプションを含めると、付与対象者は、ユーザーが付与対象者に付与する列に対する許可のみを付与できます。

> **Note**
>
> また、列フィルターは、列フィルターを指定し、すべての行を行フィルターとして指定するデータフィルターを作成することによってのみ、適用できます。ただし、これには追加の手順が必要になります。

- 列、行、またはセルのフィルタリングを適用します。

  1. [Advanced cell-level filters] (高度なセルレベルのフィルター) を選択します。



  2. (オプション) [View existing permissions] (既存の許可を表示) を展開します。
  3. (オプション) [Create new filter] (新しいフィルターを作成) を選択します。
  4. (オプション) リストされたフィルターの詳細を表示する、または新しいフィルターの作成や既存のフィルターの削除を実行するには、[Manage filters] (フィルターを管理) を選択します。

     [Data filters] (データフィルター) ページは、新しいブラウザで開きます。

     [Data filters] (データフィルター) ページでの作業を終えたら、[Grant permissions] (許可の付与) ページに戻り、必要に応じてページを更新して、作成した新しいデータフィルターを表示します。
  5. この付与に適用する 1 つ、または複数のデータフィルターを選択します。

> **Note**
>
> リストにデータフィルターがない場合は、選択したテーブルに対してデータフィルターが作成されていないことを意味します。

8. [Grant] (付与) を選択します。

AWS CLI

テーブル許可は、名前付きリソース方式と AWS Command Line Interface (AWS CLI) を使用して付与することができます。

を使用してテーブルのアクセス許可を付与するには AWS CLI

- `grant-permissions` コマンドを実行し、リソースとしてテーブルを指定します。

Example– 単一のテーブルに対する付与 – フィルタリングなし

次の例ではALTER、データベース `datalake_user1`の テーブルの AWS アカウント 1111-2222-3333 `inventory`のユーザーに SELECTと を付与しますretail。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "SELECT" "ALTER" --resource '{ "Table": {"DatabaseName":"retail",
 "Name":"inventory"}}'
```

> **Note**
>
> 登録された口ケーションに基盤となるデータを持つテーブルに対する ALTER 許可を付与する場合は、その口ケーションに対するデータ口ケーション許可もプリンシパルに付与するようにしてください。詳細については、「データ口ケーション許可の付与」を参照してください。

Example– 付与オプションを使用したすべてのテーブルに対する付与 – フィルタリングなし

次の例は、データベース `retail` 内のすべてのテーブルに対する grant オプション付きの SELECT を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "SELECT" --permissions-with-grant-option "SELECT" --resource '{ "Table":
 { "DatabaseName": "retail", "TableWildcard": {} } }'
```

Example– シンプルな列フィルタリングを使用する付与

次の例は、表 persons 内の列のサブセットに対する SELECT を付与します。これは、シンプルな列フィルタリングを使用します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "SELECT" --resource '{ "TableWithColumns": {"DatabaseName":"hr",
 "Name":"persons", "ColumnNames":["family_name", "given_name", "gender"]}}'
```

Example– データフィルターを適用する付与

この例は、orders テーブルに対する SELECT を付与し、restrict-pharma データフィルターを適用します。

```
aws lakeformation grant-permissions --cli-input-json file://grant-params.json
```

以下は、ファイル grant-params.json の内容です。

```
{
    "Principal": {"DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
datalake_user1"},
    "Resource": {
        "DataCellsFilter": {
            "TableCatalogId": "111122223333",
            "DatabaseName": "sales",
            "TableName": "orders",
            "Name": "restrict-pharma"
        }
    },
    "Permissions": ["SELECT"],
    "PermissionsWithGrantOption": ["SELECT"]
}
```

関連情報

- Lake Formation 許可の概要
- Lake Formation でのデータフィルタリングとセルレベルのセキュリティ
- Lake Formation のペルソナと IAM 許可のリファレンス

- リソースリンク許可の付与
- 共有 Data Catalog テーブルとデータベースへのアクセスと表示

名前付きリソース方式を使用したビューに対するアクセス権限の付与

以下の手順では、名前付きリソース方式と、「アクセス許可の付与」ページを使用して、ビューに対するアクセス許可を付与する方法を説明します。このページは、以下のセクションに分かれています。

- プリンシパルタイプ – アクセス許可を付与する IAM ユーザー、ロール、IAM Identity Center ユーザーとグループ AWS アカウント、組織、または組織単位。一致する属性を持つプリンシパルにアクセス許可を付与することもできます。
- [LF タグまたはカタログリソース] – 付与する許可の対象となるデータベース、テーブル、ビュー、またはリソースリンク。
- [許可] – 付与されるデータレイク許可。

「アクセス許可の付与」ページを開きます。

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、データレイク管理者、データベース作成者、またはデータベースに対する付与可能なアクセス許可を持つ IAM ユーザーとしてサインインします。
2. 次のいずれかを行います。
   - ナビゲーションペインの [Permissions] (許可) で [Data permissions] (データの許可) を選択します。次に、[Grant] (付与) を選択します。
   - ナビゲーションペインの [データカタログ] で、[ビュー] を選択します。次に、[ビュー] ページでビューを選択し、[アクション] メニューの [許可] で [付与] を選択します。

> **Note**
>
> ビューに対する許可は、リソースリンクを使用して付与できます。これを実行するには、[ビュー] ページでリソースリンクを選択し、[アクション] メニューで [ターゲットに対して付与] を選択します。詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。

プリンシパルタイプを指定する

[プリンシパルタイプ] セクションで、プリンシパルまたは属性別のプリンシパルを選択します。プリンシパルを選択した場合、次のオプションを使用できます。

IAM ユーザーとロール

[IAM users and roles] (IAM ユーザーおよびロール) リストから、1 人、または複数のユーザーまたはロールを選択します。

IAM アイデンティティセンター

[ユーザーとグループ] リストから、1 人、または複数のユーザーまたはグループを選択します。

SAML ユーザーとグループ

SAML および Quick ユーザーおよびグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、または Amazon Quick ユーザーまたはグループの ARNs を入力します。各 ARN の後で Enter キーを押します。

ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。

> **Note**
>
> Lake Formation と Quick の統合は、Quick Enterprise Edition でのみサポートされています。

外部アカウント

AWS アカウント、 AWS 組織、または IAM プリンシパルには、IAM ユーザーまたはロールの 1 つ以上の有効な AWS アカウント IDs、組織 IDs、組織単位 IDs、または ARN を入力します。各 ID の後で [Enter] キーを押します。

組織 ID は、最初の「o-」と、その後に続く 10 ~ 32 個の小文字または数字で構成されています。

組織単位 ID は「ou-」で始まり、その後に 4 ~ 32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」ダッシュと 8 ~ 32 個の追加の小文字または数字が続きます。

以下の資料も参照してください。

- 共有 Data Catalog テーブルとデータベースへのアクセスと表示

属性別のプリンシパル

属性のキーと値を指定します。複数の値を選択する場合は、OR 演算子で属性式を作成することになります。つまり、IAM ロールまたはユーザーに割り当てられた属性タグ値のいずれかが一致した場合、ロール/ユーザーはリソースに対するアクセス許可を取得します。

同じアカウントまたは別のアカウントで一致する属性を持つプリンシパルにアクセス許可を付与するかどうかを指定して、アクセス許可の範囲を選択します。

ビューを指定します。

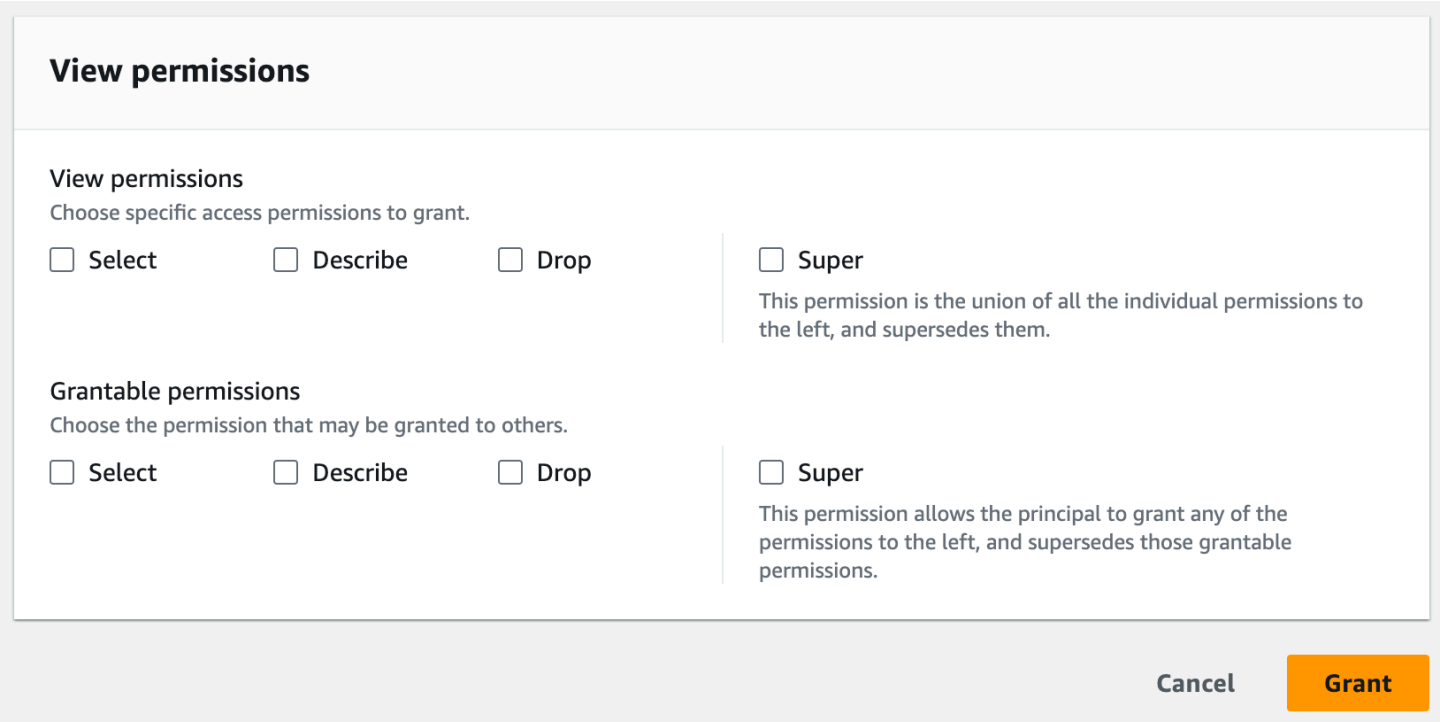
[LF タグまたはカタログリソース] セクションで、付与する許可の対象となるビューを 1 つ、または複数選択します。

1. [Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。
2. [ビュー] リストから 1 つまたは複数のビューを選択します。1 つ以上のカタログ、データベース、テーブル、データフィルターを選択することもできます。

   データベース内の `All tables` にデータレイクのアクセス許可を付与すると、被付与者はデータベース内のすべてのテーブルとビューに対するアクセス許可を持つことになります。

許可を指定する

[Permissions] (許可) セクションで、許可と付与可能な許可を選択します。



1. [アクセス許可の表示] で、付与する許可を 1 つ、または複数選択します。
2. (オプション) [Grantable permissions] (付与可能な許可) で、付与対象者がそれぞれの AWS アカウント内の他のプリンシパルに付与できる許可を選択します。このオプションは、外部アカウントから IAM プリンシパルにアクセス許可を付与する場合はサポートされません。
3. [Grant] (付与) を選択します。

以下の資料も参照してください。

- Lake Formation 許可のリファレンス
- アカウントと共有されたデータベースまたはテーブルに対する許可の付与

Lake Formation のタグベースのアクセス制御

Lake Formation のタグベースのアクセス制御 (LF-TBAC) は、属性に基づいて許可を定義する認可戦略です。これらの属性は、Lake Formation で LF タグと呼ばれています。データカタログリソースに LF タグをアタッチし、それらの LF タグを使用して、リソースに対するアクセス許可を Lake Formation プリンシパルに付与できます。Lake Formation は、プリンシパルでリソースのタグ値と一

致するタグ値にアクセス許可が付与されたときに、それらのリソースに対するオペレーションを許可します。

LF-TBAC は、急成長する環境や、ポリシー管理が煩雑になる状況で役に立ちます。

LF-TBAC は、データカタログオブジェクト (フェデレーティッドカタログ、データベース、テーブル、ビューなど) が多数ある場合に Lake Formation 許可を付与するために使用することが推奨される方式です。Lake Formation は、Amazon S3 テーブル、 Amazon Redshift データウェアハウス、SQL Server Amazon DynamoDB、Snowflake などのフェデレーティッドデータソースのフェデレーティッドカタログのタグベースのアクセスコントロールをサポートしています。

> **Note**
>
> IAM タグと LF タグは同じではありません。これらのタグは置き換え可能ではありません。LF タグは Lake Formation アクセス許可を付与するために使用され、IAM タグは IAM ポリシーを定義するために使用されます。

Lake Formation のタグベースのアクセス制御の仕組み

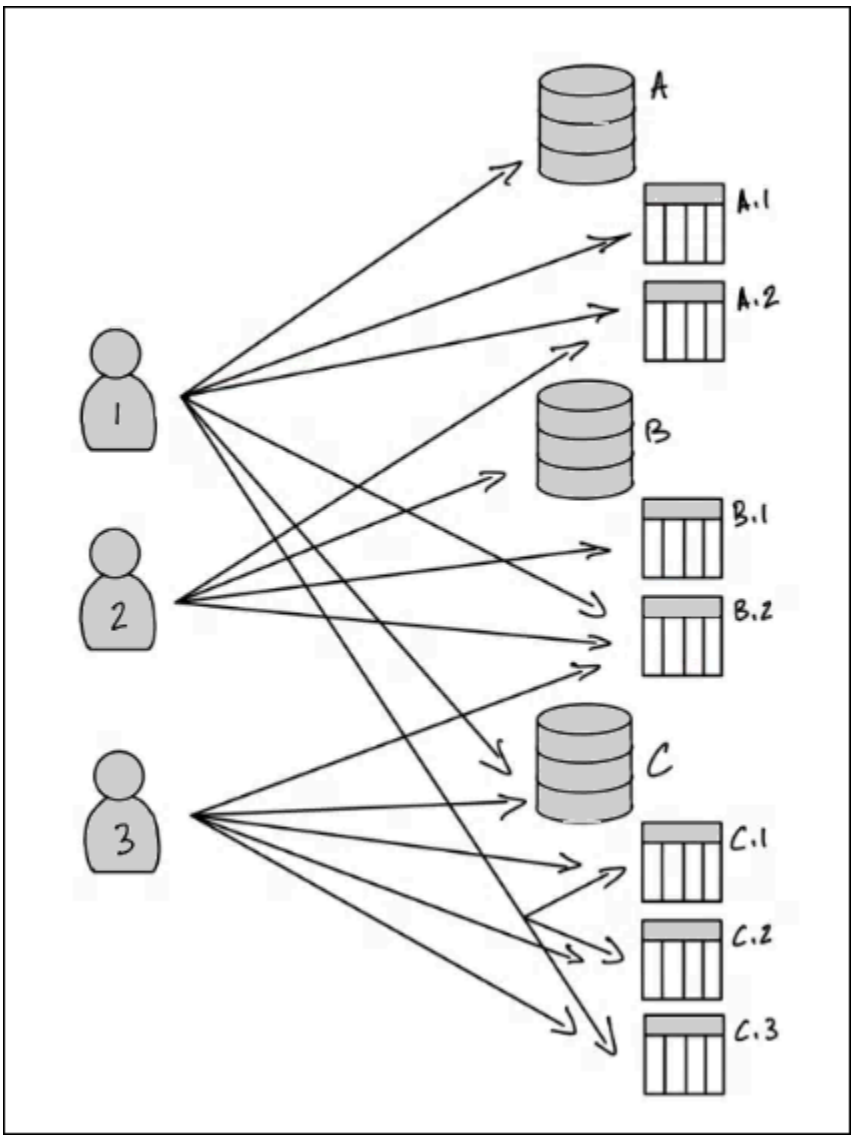
各 LF タグは、`department=sales` や `classification=restricted` などのキーと値のペアです。キーは、`department=sales,marketing,engineering,finance` など複数の定義された値を持つことができます。

LF-TBAC 方式を使用するには、データレイク管理者とデータエンジニアが以下のタスクを実行します。

タスク	タスクの詳細
1. LF タグのプロパティと関係を定義します。	-
2. Lake Formation で LF タグ作成者を作成します。	LF タグ作成者の追加
3. Lake Formation で LF タグを作成します。	LF タグの作成

タスク	タスクの詳細
4. LF タグを Data Catalog リソースに割り当てます。	Data Catalog リソースへの LF タグの割り当て
5. LF タグをリソースに割り当てる許可 (オプションで付与オプションを使用) を他のプリンシパルに付与します。	LF タグ値のアクセス許可の管理
6. LF タグ式 (オプションで付与オプションを使用) をプリンシパルに付与します。	LF-TBAC 方式を使用したデータレイク許可の付与
7. (推奨) プリンシパルが LF-TBAC 方式を使用して正しいリソースにアクセスできることを確認した後、名前付きリソース方式を使用して付与された許可を取り消します。	-

3 つのデータベースと 7 つのテーブルに対するアクセス許可を 3 人のプリンシパルに付与する必要がある場合を考えてみましょう。



上の図に示されているアクセス許可を名前付きリソース方法を使用して実現するには、以下のように、17 の付与を行う必要があります (擬似コードを使用)。

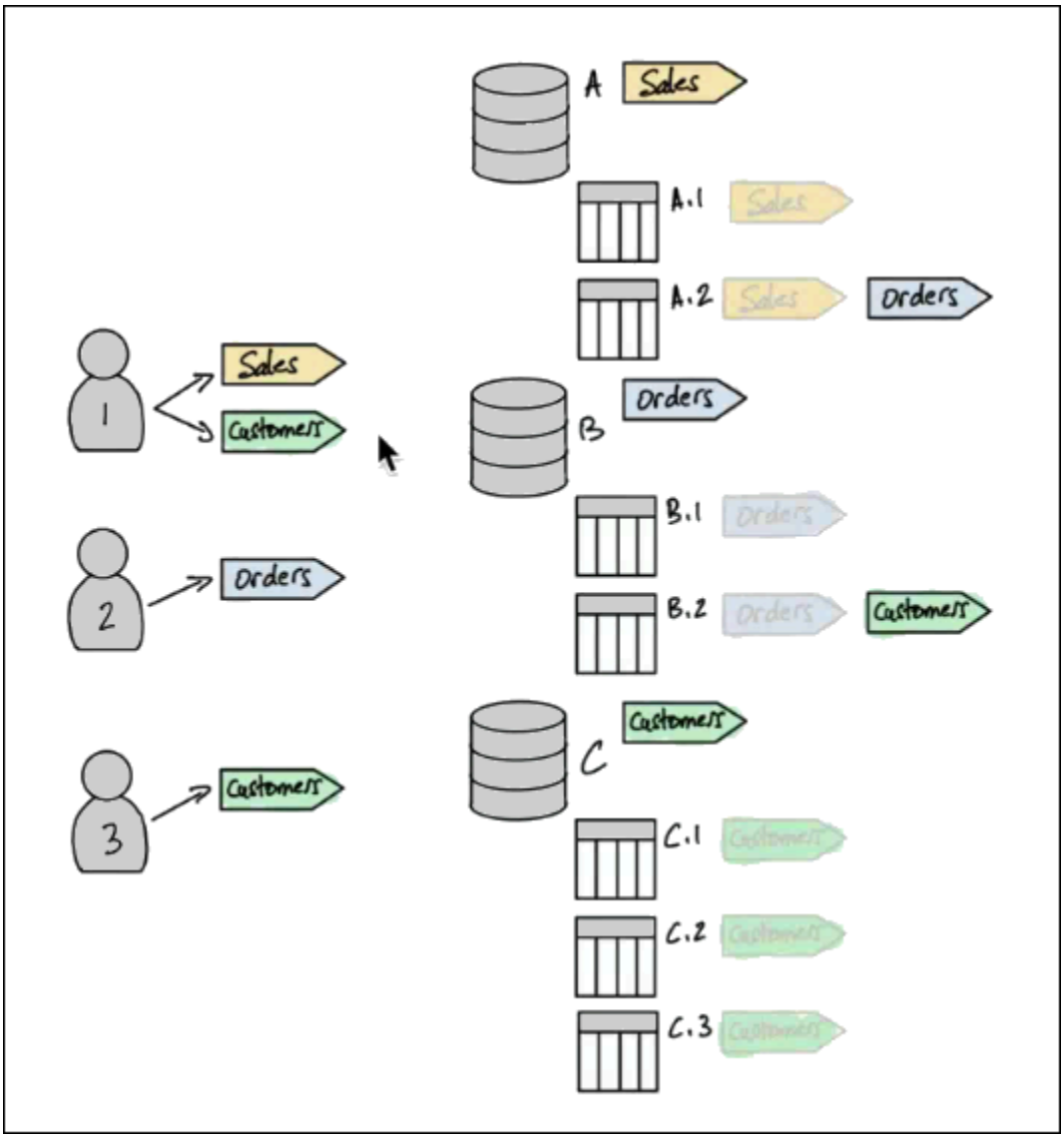
```
GRANT CREATE_TABLE ON Database A TO PRINCIPAL 1
GRANT SELECT, INSERT ON Table A.1 TO PRINCIPAL 1
GRANT SELECT, INSERT ON Table A.2 TO PRINCIPAL 1
GRANT SELECT, INSERT ON Table B.2 TO PRINCIPAL 1
...
GRANT SELECT, INSERT ON Table A.2 TO PRINCIPAL 2
GRANT CREATE_TABLE ON Database B TO PRINCIPAL 2
...
GRANT SELECT, INSERT ON Table C.3 TO PRINCIPAL 3
```

今度は、LF-TBAC を使用してアクセス許可を付与する方法を考えてみます。次の図は、LF タグをデータベースとテーブルに割り当てて、LF タグに対するアクセス許可をプリンシパルに付与したことを示しています。

この例では、LF タグが、エンタープライズリソースプランニング (ERP) アプリケーションスイートの異なるモジュールの分析が含まれるデータレイクの領域を表しています。さまざまなモジュールの分析データへのアクセスを制御するために使用できます。すべての LF タグは、`module` というキーと、`Sales`、`Orders`、および `Customers` の可能な値を持っています。LF タグの例は以下のようになります。

```
module=Sales
```

この図は LF タグの値のみを示しています。



Data Catalog リソースへのタグ割り当てと継承

テーブルはデータベースから LF タグを継承し、列はテーブルから LF タグを継承します。継承された値は上書きすることができます。上記の図では、淡色表示の LF タグが継承されています。

継承が行われるため、データレイク管理者は、リソースに対して以下の 5 つの LF タグの割り当てを行うだけで済みます (擬似コードを使用)。

```
ASSIGN TAGS module=Sales TO database A
ASSIGN TAGS module=Orders TO table A.2
ASSIGN TAGS module=Orders TO database B
ASSIGN TAGS module=Customers TO table B.2
ASSIGN TAGS module=Customers TO database C
```

プリンシパルへのタグの付与

データベースとテーブルに LF タグを割り当てた後、データレイク管理者は、以下のようにプリンシパルに対して LF タグを 4 回付与するだけで済みます (擬似コードを使用)。

```
GRANT TAGS module=Sales TO Principal 1
GRANT TAGS module=Customers TO Principal 1
GRANT TAGS module=Orders TO Principal 2
GRANT TAGS module=Customers TO Principal 3
```

これで、LF タグ `module=Sales` を持つプリンシパルは LF タグ `module=Sales` を持つ Data Catalog リソース (例えば、データベース A) にアクセスでき、LF タグ `module=Customers` を持つプリンシパルは LF タグ `module=Customers` を持つリソースにアクセスできる、というようになります。

上記の grant コマンドは不完全です。これらは、プリンシパルが許可を持つ Data Catalog リソースを LF タグで示してはいるものの、プリンシパルがそれらのリソースに対してどの Lake Formation 許可 (SELECT、ALTER など) を持っているかを正確に示していないためです。したがって、以下の擬似コードのコマンドが、LF タグを使用して Data Catalog リソースに対する Lake Formation 許可を付与する方法のより正確な表現になります。

```
GRANT (CREATE_TABLE ON DATABASES) ON TAGS module=Sales TO Principal 1
GRANT (SELECT, INSERT ON TABLES)  ON TAGS module=Sales TO Principal 1
GRANT (CREATE_TABLE ON DATABASES) ON TAGS module=Customers TO Principal 1
GRANT (SELECT, INSERT ON TABLES)  ON TAGS module=Customers TO Principal 1
GRANT (CREATE_TABLE ON DATABASES) ON TAGS module=Orders TO Principal 2
```

```
GRANT (SELECT, INSERT ON TABLES)  ON TAGS module=Orders TO Principal 2
GRANT (CREATE_TABLE ON DATABASES) ON TAGS module=Customers TO Principal 3
GRANT (SELECT, INSERT ON TABLES)  ON TAGS module=Customers TO Principal 3
```

まとめ – 結果として得られたリソースに対するアクセス許可

以下の表は、上記の図のデータベースとテーブルに割り当てられた LF タグと、図の中でプリンシパルに付与された LF タグを前提とした、プリンシパルが持つデータベースとテーブルに対する Lake Formation 許可のリストです。

プリンシパル	LF タグを通じて付与された許可
プリンシパル 1	• データベース A に対する CREATE_TABLE • テーブル A.1 に対する SELECT、INSERT • テーブル B.2 に対する SELECT、INSERT • データベース C に対する CREATE_TABLE • テーブル C.1 に対する SELECT、INSERT • テーブル C.2 に対する SELECT、INSERT • テーブル C.3 に対する SELECT、INSERT
プリンシパル 2	• テーブル A.2 に対する SELECT、INSERT • データベース B に対する CREATE_TABLE • テーブル B.1 に対する SELECT、INSERT
プリンシパル 3	• テーブル B.2 に対する SELECT、INSERT • データベース C に対する CREATE_TABLE • テーブル C.1 に対する SELECT、INSERT • テーブル C.2 に対する SELECT、INSERT • テーブル C.3 に対する SELECT、INSERT

結論

このシンプルな例では、5 つの割り当て操作と 8 つの付与操作を使用することで、データレイク管理者が 17 個の許可を指定できました。何十個ものデータベースと、数百個ものテーブルがあるときは、名前付きリソース方式に勝る LF-TBAC 方式の利点が明白になります。すべてのプリンシパル

にすべてのリソースへのアクセス権を付与する必要があり、n(P) をプリンシパルの数、n(R) をリソースの数とする仮定上のケースでは、以下のようになります。

- 名前付きリソース方式では、必要な付与数が n(P) × n(R) 個になります。
- 単一の LF タグを使用する LF-TBAC 方式では、プリンシパルへの付与とリソースへの割り当ての合計数が n(P) + n(R) 個になります。

> **関連情報**
>
> - メタデータアクセスコントロールのための LF タグの管理
> - LF-TBAC 方式を使用したデータレイク許可の付与

トピック

- メタデータアクセスコントロールのための LF タグの管理
- メタデータアクセスコントロールのための LF タグ式の管理
- LF タグ値のアクセス許可の管理

メタデータアクセスコントロールのための LF タグの管理

Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方式を使用して、データカタログオブジェクト (カタログ、データベース、テーブル、および列) をセキュリティで保護するには、LF タグを作成し、それらをリソースに割り当てて、LF タグアクセス許可をプリンシパルに付与します。

LF タグをデータカタログオブジェクトに割り当てたり、アクセス許可をプリンシパルに付与したりする前に、LF タグを定義する必要があります。LF タグを作成できるのは、データレイク管理者または LF タグ作成者アクセス許可を持つプリンシパルのみです。

LF タグ作成者

LF タグ作成者は、LF タグを作成および管理するアクセス許可を持つ非管理者プリンシパルです。データレイク管理者は、Lake Formation コンソールまたは CLI を使用して LF タグ作成者を追加できます。LF タグ作成者には、LF タグを更新および削除したり、LF タグをリソースに割り当てたり、他のプリンシパルに LF タグアクセス許可と LF タグ値アクセス許可を付与したりするための暗黙の Lake Formation アクセス許可があります。

LF タグ作成者のロールにより、データレイク管理者はタグキーや値の作成や更新などのタグ管理タスクを管理者以外のプリンシパルに委任できます。データレイク管理者は LF タグ作成者に付与可能な `Create LF-Tag` アクセス許可を付与することもできます。その後、LF タグ作成者は、LF タグを作成するアクセス許可を他のプリンシパルに付与できます。

LF タグに対する次の 2 種類のアクセス許可を付与できます。

- LF タグアクセス許可 - `Create LF-Tag`、`Alter`、および `Drop`。これらのアクセス許可は、LF タグの作成、更新、および削除に必要です。

  データレイク管理者と LF タグ作成者は、作成した LF タグに対するこれらのアクセス許可を暗黙的に持ち、これらのアクセス許可をプリンシパルに明示的に付与し、データレイク内のタグを管理できます。
- LF タグのキーと値のペアのアクセス許可 - `Assign`、`Describe`、および `Grant with LF-Tag expressions`。これらのアクセス許可は、LF タグをデータカタログオブジェクトに割り当てたり、Lake Formation タグベースのアクセスコントロールを使用してリソースに対するアクセス許可をプリンシパルに付与したりするために必要です。LF タグ作成者は、LF タグを作成するときに、これらのアクセス許可を暗黙的に受け取ります。

`Create LF-Tag` アクセス許可を受け取り、LF タグの作成に成功すると、LF タグ作成者は、LF タグをリソースに割り当てて、LF タグアクセス許可 (`Create LF-Tag`、`Alter`、`Drop`) を管理者以外の他のプリンシパルに付与して、データレイク内のタグを管理できます。Lake Formation コンソール、API、または AWS Command Line Interface () を使用して LF タグを管理できますAWS CLI。

> **Note**
>
> データレイク管理者は、LF タグの作成、更新、削除、LF タグのリソースへの割り当て、および LF タグアクセス許可のプリンシパルへの付与を行う暗黙的な Lake Formation アクセス許可を持っています。

ベストプラクティスと考慮事項については、「Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項」を参照してください。

トピック

- LF タグ作成者の追加
- LF タグの作成

- LF タグの更新
- LF タグの削除
- LF タグのリスト化
- Data Catalog リソースへの LF タグの割り当て
- リソースに割り当てられた LF タグの表示
- LF タグが割り当てられているリソースの表示
- LF タグのライフサイクル
- Lake Formation のタグベースのアクセス制御と IAM の属性ベースのアクセス制御の比較

> **関連情報**
>
> - LF タグ値のアクセス許可の管理
> - LF-TBAC 方式を使用したデータレイク許可の付与
> - Lake Formation のタグベースのアクセス制御

LF タグ作成者の追加

デフォルトでは、データレイク管理者は、LF タグの作成、更新、削除、データカタログオブジェクトへのタグの割り当て、プリンシパルへのタグアクセス許可の付与を行うことができます。タグの作成および管理操作を管理者以外のプリンシパルに委任する場合、データレイク管理者は LF タグ作成者ロールを作成して、Lake Formation `Create LF-Tag` アクセス許可をロールに付与することができます。付与可能な `Create LF-Tag` アクセス許可がある場合、LF タグ作成者は、タグの作成およびメンテナンスタスクを管理者以外の他のプリンシパルに委任できます。

データレイク管理者が LF タグをデータカタログリソースに割り当てるには、自身が作成したものではない LF タグに対する関連付けアクセス許可を、自身に付与することが必要になります。

> **Note**
>
> クロスアカウントアクセス許可の付与には、`Describe` および `Associate` アクセス許可のみを含めることができます。`Create LF-Tag`、`Drop`、`Alter`、および `Grant with LFTag expressions` アクセス許可を別のアカウントのプリンシパルに付与することはできません。

トピック

- LF タグの作成に必要な IAM アクセス許可
- LF タグ作成者の追加

関連情報

- LF タグ値のアクセス許可の管理
- LF-TBAC 方式を使用したデータレイク許可の付与
- Lake Formation のタグベースのアクセス制御

LF タグの作成に必要な IAM アクセス許可

Lake Formation のプリンシパルが LF タグを作成できるようにアクセス許可を設定する必要があります。LF タグ作成者になる必要があるプリンシパルのアクセス許可ポリシーに、以下のステートメントを追加します。

Note

データレイク管理者は、LF タグの作成、更新、削除、LF タグのリソースへの割り当て、および LF タグのプリンシパルへの付与を行う暗黙的な Lake Formation アクセス許可を持っていますが、データレイク管理者には以下の IAM アクセス許可も必要です。

詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

```
{
"Sid": "Transformational",
"Effect": "Allow",
    "Action": [
        "lakeformation:AddLFTagsToResource",
        "lakeformation:RemoveLFTagsFromResource",
        "lakeformation:GetResourceLFTags",
        "lakeformation:ListLFTags",
        "lakeformation:CreateLFTag",
        "lakeformation:GetLFTag",
        "lakeformation:UpdateLFTag",
        "lakeformation:DeleteLFTag",
```

```
        "lakeformation:SearchTablesByLFTags",
        "lakeformation:SearchDatabasesByLFTags"
    ]
}
```

リソースに LF タグを付与し、プリンシパルに LF タグを付与するプリンシパルは、CreateLFTag、UpdateLFTag、および DeleteLFTag 許可を除き、同じ許可を持っている必要があります。

LF タグ作成者の追加

LF タグ作成者は、LF タグの作成、タグのキーと値の更新、タグの削除、データカタログリソースへのタグの関連付け、および LF-TBAC 方法を使用して、プリンシパルへのデータカタログリソースに対するアクセス許可の付与を行うことができます。LF タグ作成者は、これらのアクセス許可をプリンシパルに付与することもできます。

LF タグ作成者ロールを作成するには、 AWS Lake Formation コンソール、 API、または AWS Command Line Interface () を使用しますAWS CLI。

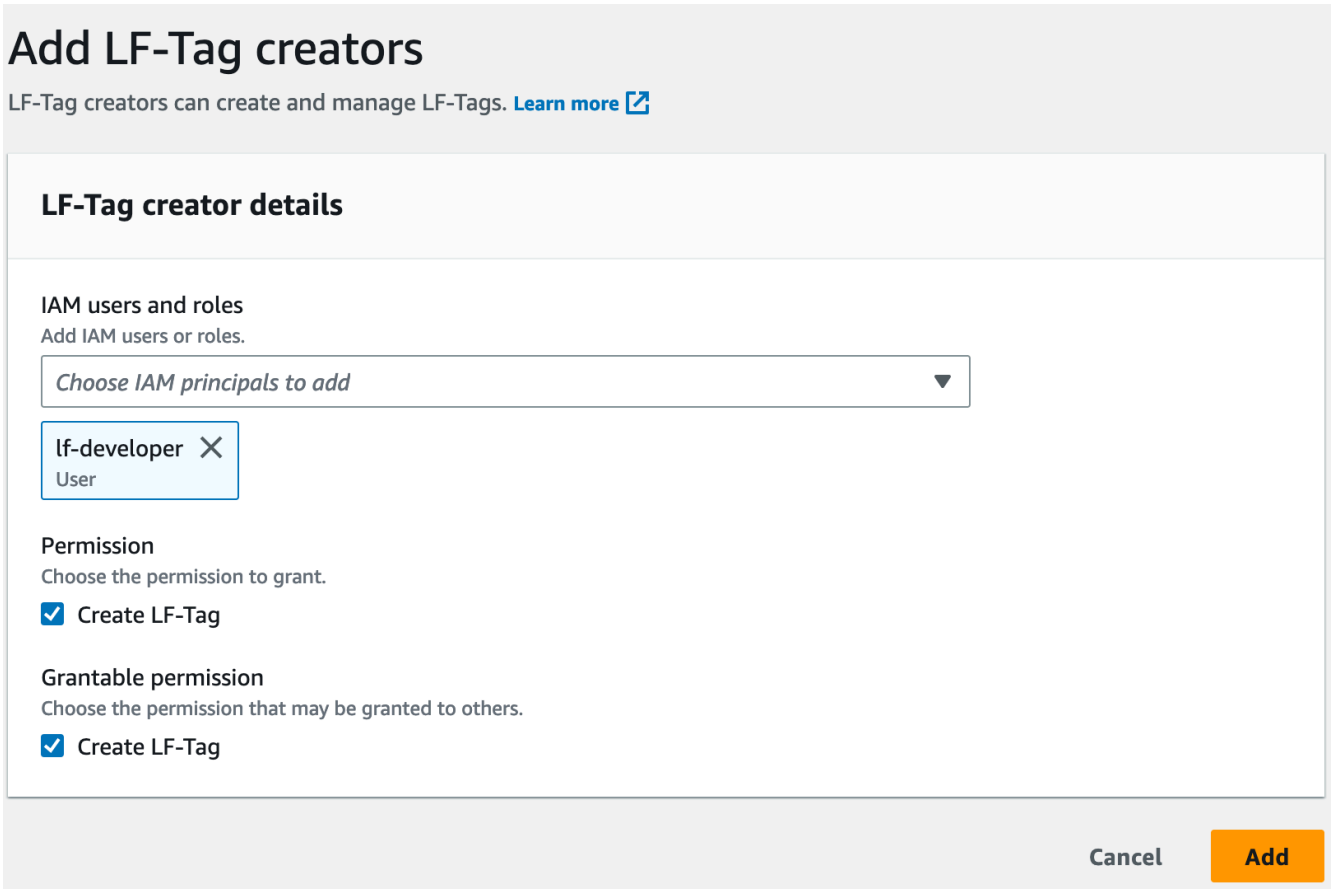
console

LF タグ作成者を追加するには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者としてサインインします。
2. ナビゲーションペインで、[アクセス許可] の [LF タグとアクセス許可] を選択します。

   [LF タグとアクセス許可] ページで、[LF タグ作成者] セクションを選択し、[LF タグ作成者の追加] を選択します。



3. [LF タグ作成者の追加] ページで、LF タグの作成に必要なアクセス許可を持つ IAM ロールまたはユーザーを選択します。
4. [Create LF-Tag アクセス許可] チェックボックスをオンにします。
5. (オプション) 選択したプリンシパルが Create LF-Tag アクセス許可をプリンシパルに付与できるようにするには、[付与可能な Create LF-Tag アクセス許可] を選択します。
6. [Add] (追加) を選択します。

AWS CLI

```
aws lakeformation grant-permissions --cli-input-json file://grantCreate
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:user/tag-manager"
    },
    "Resource": {
        "Catalog": {}
    },
    "Permissions": [
```

```
        "CreateLFTag"
    ],
    "PermissionsWithGrantOption": [
        "CreateLFTag"
    ]
}
```

LF タグ作成者ロールで利用できるアクセス許可は次のとおりです。

アクセス許可	説明
Drop	LF タグに対するこのアクセス許可を持つプリンシパルは、データレイクから LF タグを削除できます。プリンシパルは、LF タグリソースのすべてのタグ値に対する暗黙的な Describe アクセス許可を取得します。
Alter	LF タグに対するこのアクセス許可を持つプリンシパルは、LF タグにタグ値を追加したり、LF タグからタグ値を削除したりできます。プリンシパルは、LF タグリソースのすべてのタグ値に対する暗黙的な Alter アクセス許可を取得します。
Describe	LF タグに対するこのアクセス許可を持つプリンシパルは、LF タグをリソースに割り当てるとき、または LF タグに対するアクセス許可を付与するときに、LF タグとその値を表示できます。すべてのキーの値、または特定のキーの値に対する Describe を付与することができます。
Associate	LF タグに対してこの許可を持つプリンシパルは、LF タグを Data Catalog リソースに割り当てることができます。Associate の付与は、Describe を黙示的に付与します。
Grant with LF-Tag expression	LF タグに対するこのアクセス許可を持つプリンシパルは、LF タグのキーと値を使用して、データカタログリソースに対するアクセス許可を付与できます。Grant with LF-Tag expression の付与は、Describe を黙示的に付与します。

これらの許可は付与可能です。これらの許可を grant オプションと共に付与されたプリンシパルは、これらを他のプリンシパルに付与できます。

LF タグの作成

すべての LF タグは、使用前に Lake Formation で定義される必要があります。LF タグは、キーと、キーに対する 1 つ以上の可能な値で構成されます。

データレイク管理者が LF タグ作成者ロールに必要な IAM アクセス許可と Lake Formation アクセス許可を設定したら、プリンシパルは LF タグを作成できます。LF タグ作成者は、LF タグの任意のタグ値を更新または削除したり、LF タグを削除したりする暗黙的なアクセス許可を取得します。

LF タグは、 AWS Lake Formation コンソール、 API、または AWS Command Line Interface () を使用して作成できますAWS CLI。

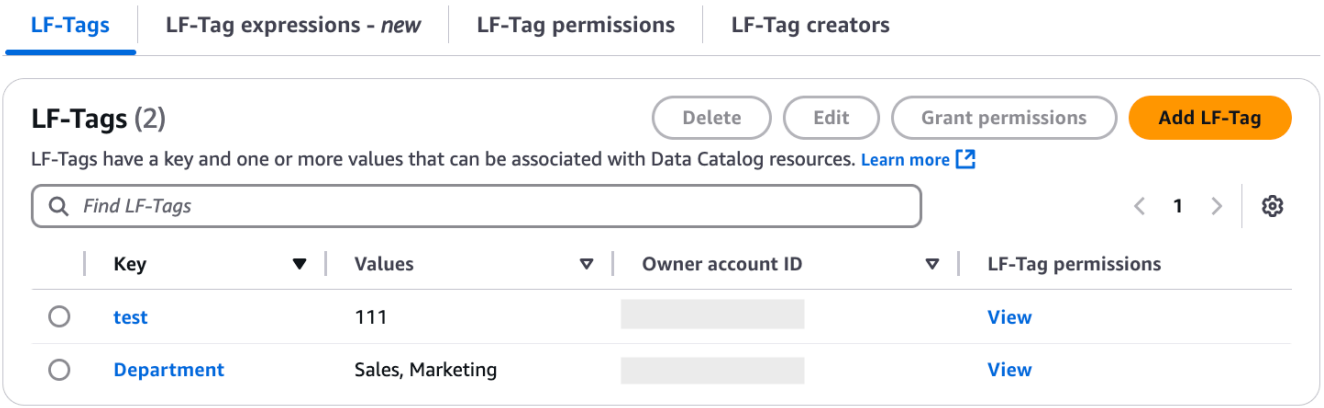
Console

LF タグを作成するには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   LF タグ作成者アクセス許可を持つプリンシパルまたはデータレイク管理者としてサインインします。
2. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] の [LF タグ] を選択します。

   [LF-Tags] (LF タグ) ページが表示されます。



3. [Add LF-Tag] (LF タグを追加) を選択します。
4. [Add LF-Tag] (LF タグの追加) ダイアログボックスで、キーと、1 つまたは複数の値を入力します。

各キーには、少なくとも 1 つの値が必要です。複数の値を入力するには、カンマ区切りのリストを入力してから [Enter] キーを押すか、一度に 1 つの値を入力し、入力するたびに [Add] (追加) を選択します。許可される値の最大数は 1000 です。

5. [Add tag] (タグを追加) を選択します。

AWS CLI

LF タグを作成するには

- `create-lf-tag` コマンドを入力します。

  次の例は、キー `module` と値 `Customers` および `Orders` を持つ LF タグを作成します。

```
aws lakeformation create-lf-tag --tag-key module --tag-values Customers Orders
```

タグ作成者になると、プリンシパルは、この LF タグに対する `Alter` アクセス許可を取得し、この LF タグの任意のタグ値を更新または削除できます。LF タグ作成者プリンシパルは、この LF タグのタグ値を更新および削除する `Alter` アクセス許可を他のプリンシパルに付与することもできます。

LF タグの更新

`Alter` アクセス許可がある LF タグを更新するには、許可されたキー値を追加または削除します。LF タグのキーを変更することはできません。キーを変更するには、LF タグを削除して、必要なキーを持つ LF タグを追加します。値を更新するには、`Alter` アクセス許可のほかに、`lakeformation:UpdateLFTag` IAM アクセス許可も必要です。

LF タグの値を削除するときには、データカタログリソースにその LF タグの値が存在するかどうかのチェックは実行されません。削除された LF タグの値がリソースに関連付けられていた場合、この値はそのリソースで認識されなくなり、そのキーと値のペアに対するアクセス許可が付与されたプリンシパルはアクセス許可を失います。

LF タグの値を削除する前に、オプションで `remove-lf-tags-from-resource` コマンドを使用して、削除する値があるデータカタログリソースから LF タグを削除してから、保持したい値でリソースにタグを付け直すことができます。

データレイク管理者、LF タグ作成者、および LF タグに対する `Alter` アクセス許可を持つプリンシパルのみが、LF タグを更新できます。

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して LF タグを更新できますAWS CLI。

Console

LF タグを更新する (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   データレイク管理者、LF タグ作成者、または LF タグに対する `Alter` アクセス許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] の [LF タグ] を選択します。
3. [LF タグ] ページで、LF タグを選択し、[編集] を選択します。
4. [LF タグの編集] ダイアログボックスで、LF タグの値を追加または削除します。

   複数の値を追加するには、[Values] (値) フィールドで、カンマ区切りのリストを入力して [Enter] キーを押すか、一度に 1 つの値を入力して、入力するたびに [Add] (追加) を選択します。
5. [Save] (保存) を選択します。

AWS CLI

LF タグを更新するには (AWS CLI)

- `update-lf-tag` コマンドを入力します。以下の引数の 1 つ、または両方を入力します。
  - `--tag-values-to-add`
  - `--tag-values-to-delete`

Example

次の例は、LF タグのキー `level` の値 `vp` を値 `vice-president` に置き換えます。

```
aws lakeformation update-lf-tag --tag-key level --tag-values-to-add vice-president
--tag-values-to-delete vp
```

LF タグの削除

使用されなくなった LF タグは、削除することができます。データカタログリソースに LF タグが存在するかどうかのチェックは実行されません。削除された LF タグがリソースに関連付けられていた場合、この値はそのリソースで認識されなくなり、その LF タグに対する許可が付与されていたプリンシパルはアクセス許可を失います。

LF タグを削除する前に、オプションで `remove-lf-tags-from-resource` コマンドを使用して、すべてのリソースからその LF タグを削除することができます。

データレイク管理者、LF タグ作成者、および LF タグに対する `Drop` アクセス許可を持つプリンシパルのみが、LF タグを削除できます。プリンシパルが LF タグを削除するには、`Drop` アクセス許可のほかに、`lakeformation:DeleteLFTag` IAM アクセス許可も必要です。

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して LF タグを削除できますAWS CLI。

Console

LF タグを削除するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   データレイク管理者としてサインインします。
2. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] の [LF タグ] を選択します。
3. [LF タグ] ページで、LF タグを選択し、[削除] を選択します。
4. [タグ環境を削除しますか?] ダイアログボックスで、削除を確定するには、LF タグのキー値を指定フィールドに入力し、[削除] を選択します。

AWS CLI

LF タグを削除するには (AWS CLI)

- `delete-lf-tag` コマンドを入力します。削除する LF タグのキーを指定します。

  Example

  次の例は、キー `region` を持つ LF タグを削除します。

```
aws lakeformation delete-lf-tag --tag-key region
```

LF タグのリスト化

`Describe` または `Associate` 許可を持っている LF タグをリストすることができます。LF タグの各キーと共にリストされる値は、アクセス許可を持っている値です。

LF タグ作成者には、作成した LF タグを表示する暗黙的なアクセス許可があります。

データレイク管理者は、ローカル AWS アカウントで定義されたすべての LF タグと、`Describe` および `Associate` 許可が外部アカウントからローカルアカウントに付与されたすべての LF タグを表示することができます。データレイク管理者は、すべての LF タグのすべての値を表示することができます。

AWS Lake Formation コンソール、API、または () を使用して LF タグを AWS Command Line Interface 一覧表示できますAWS CLI。

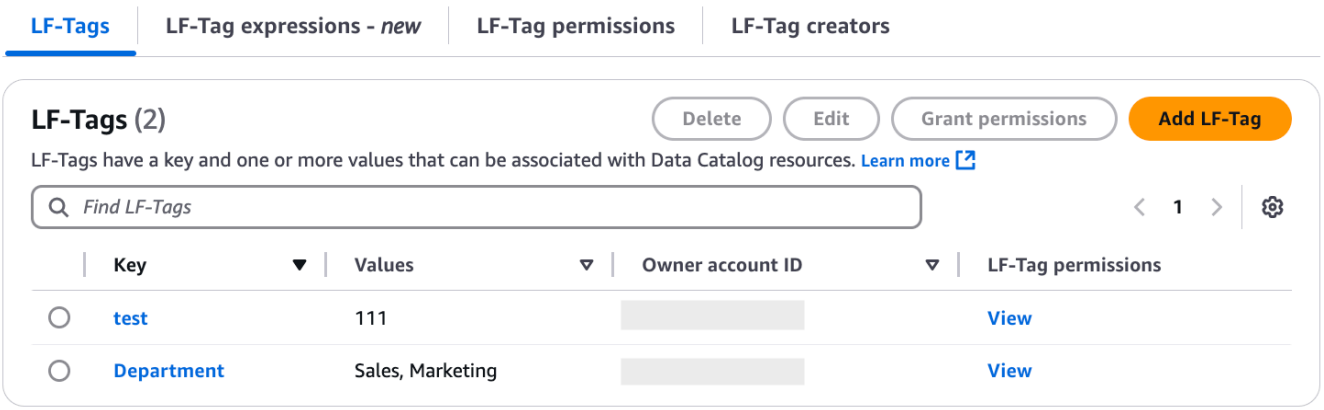
Console

LF タグをリストする (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   LF タグ作成者、データレイク管理者、または LF タグに対するアクセス許可を付与され、`lakeformation:ListLFTags` IAM 許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] の [LF タグ] を選択します。

   [LF-Tags] (LF タグ) ページが表示されます。



[所有者アカウント ID] 列をチェックして、外部アカウントからアカウントと共有された LF タグを判別します。

AWS CLI

LF タグをリストする (AWS CLI)

- 以下のコマンドを、データレイク管理者、または LF タグに対する許可を付与され、`lakeformation:ListLFTags` IAM 許可を持つプリンシパルとして実行します。

```
aws lakeformation list-lf-tags
```

出力は以下のようになります。

```
{
    "LFTags": [
        {
            "CatalogId": "111122223333",
            "TagKey": "level",
            "TagValues": [
                "director",
                "vp",
                "c-level"
            ]
        },
        {
            "CatalogId": "111122223333",
            "TagKey": "module",
            "TagValues": [
```

```
                "Orders",
                "Sales",
                "Customers"
            ]
        }
    ]
}
```

外部アカウントから付与された LF タグも表示するには、コマンドオプション `--resource-share-type ALL` を含めます。

```
aws lakeformation list-lf-tags --resource-share-type ALL
```

出力は以下のようになります。リストする LF タグがまだあることを示す NextToken キーに注意してください。

```
{
    "LFTags": [
        {
            "CatalogId": "111122223333",
            "TagKey": "level",
            "TagValues": [
                "director",
                "vp",
                "c-level"
            ]
        },
        {
            "CatalogId": "111122223333",
            "TagKey": "module",
            "TagValues": [
                "Orders",
                "Sales",
                "Customers"
            ]
        }
    ],
    "NextToken": "eyJleHBpcmF0aW...ZXh0Ijp0cnVlfQ=="
}
```

引数 --next-token を追加してコマンドを繰り返し、残りのローカル LF タグと、外部アカウントから付与された LF タグを表示します。外部アカウントからの LF タグは、常に個別のページに表示されます。

```
aws lakeformation list-lf-tags --resource-share-type ALL
--next-token eyJleHBpcmF0aW...ZXh0Ijp0cnVlfQ==
```

```
{
    "LFTags": [
        {
            "CatalogId": "123456789012",
            "TagKey": "region",
            "TagValues": [
                "central",
                "south"
            ]
        }
    ]
}
```

API

Lake Formation に利用できる SDK を使用して、リクエスト元が表示する許可を持っているタグをリストすることができます。

```
import boto3

client = boto3.client('lakeformation')
...

response = client.list_lf_tags(
    CatalogId='string',
    ResourceShareType='ALL',
    MaxResults=50'
)
```

このコマンドは、以下の構造で dict オブジェクトを返します。

```
{
    'LFTags': [
```

```
        {
            'CatalogId': 'string',
            'TagKey': 'string',
            'TagValues': [
                'string',
            ]
        },
    ],
    'NextToken': 'string'
}
```

必要な許可の詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

Data Catalog リソースへの LF タグの割り当て

LF タグを Data Catalog リソース (データベース、テーブル、および列) に割り当てて、それらのリソースへのアクセスを制御できます。リソースにアクセスできるのは、一致する LF タグが付与されたプリンシパル (および名前付きリソース方式でアクセス権が付与されたプリンシパル) のみです。

テーブルがデータベースから LF タグを継承する場合、または列がテーブルからLF タグを継承する場合は、LF タグのキーに新しい値を割り当てることで、継承された値を上書きできます。

リソースに割り当てることができる LF タグの最大数は 50 個です。

トピック

- リソースに割り当てられたタグの管理に関する要件
- LF タグをテーブル列に割り当てる
- LF タグをデータカタログリソースに割り当てる
- リソースの LF タグ の更新
- リソースからの LF タグの削除

リソースに割り当てられたタグの管理に関する要件

LF タグを Data Catalog リソースに割り当てるには、以下の要件を満たす必要があります。

- LF タグに対する Lake Formation の ASSOCIATE 許可がある。
- IAM `lakeformation:AddLFTagsToResource` の許可がある。
- Glue データベースに対する glue: GetDatabase 許可を持っている。

- リソース所有者 (作成者) である、リソースに対する GRANT オプション付きの Super Lake Formation 許可を持っている、または GRANT オプション付きの以下の許可を持っている。
  - 同じ AWS アカウントのデータベースの場合: DESCRIBE、CREATE_TABLE、ALTER、および DROP
  - 外部アカウント内のデータベースの場合: DESCRIBE、CREATE_TABLE、および ALTER
  - テーブル (および列) の場合: DESCRIBE、ALTER、DROP、INSERT、SELECT、および DELETE

さらに、LF タグとそれが割り当てられているリソースは、同じ AWS アカウントにある必要があります。

データカタログリソースから LF タグを削除するには、これらの要件を満たすとともに、lakeformation:RemoveLFTagsFromResource IAM アクセス許可も持っている必要があります。

LF タグをテーブル列に割り当てる

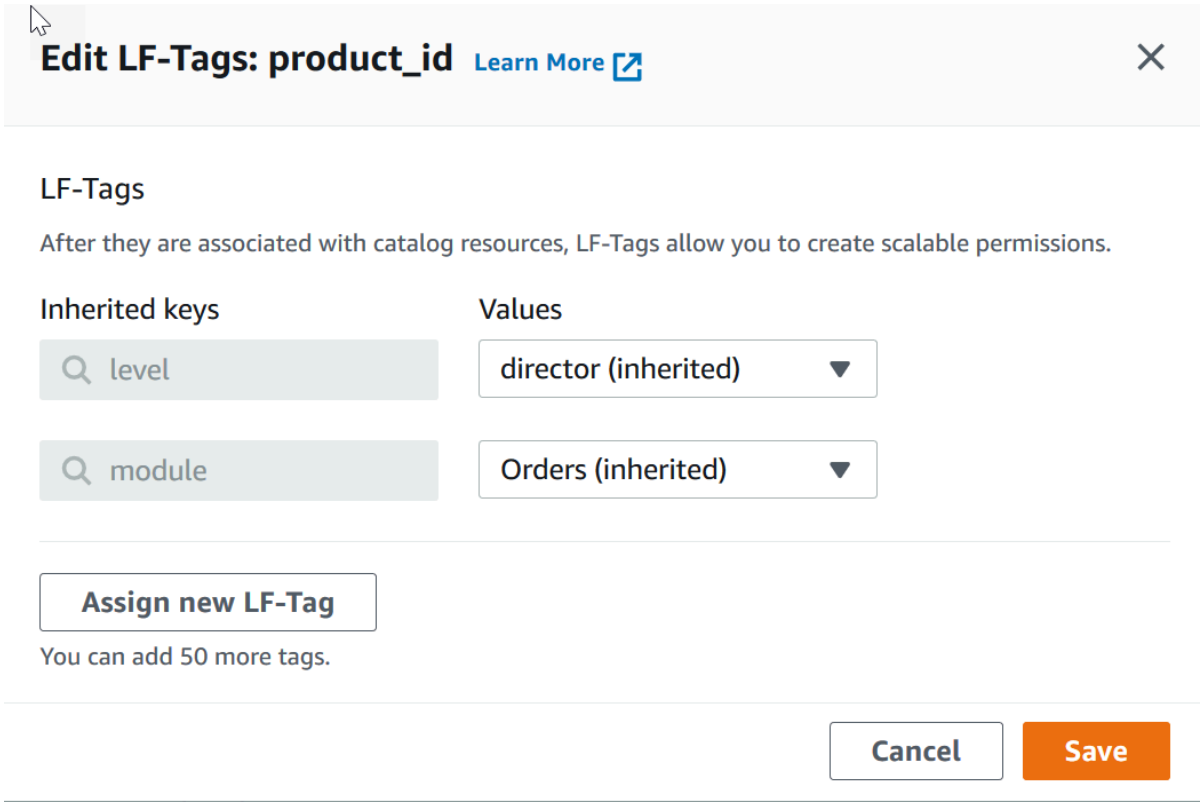
LF タグをテーブル列に割り当てるには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

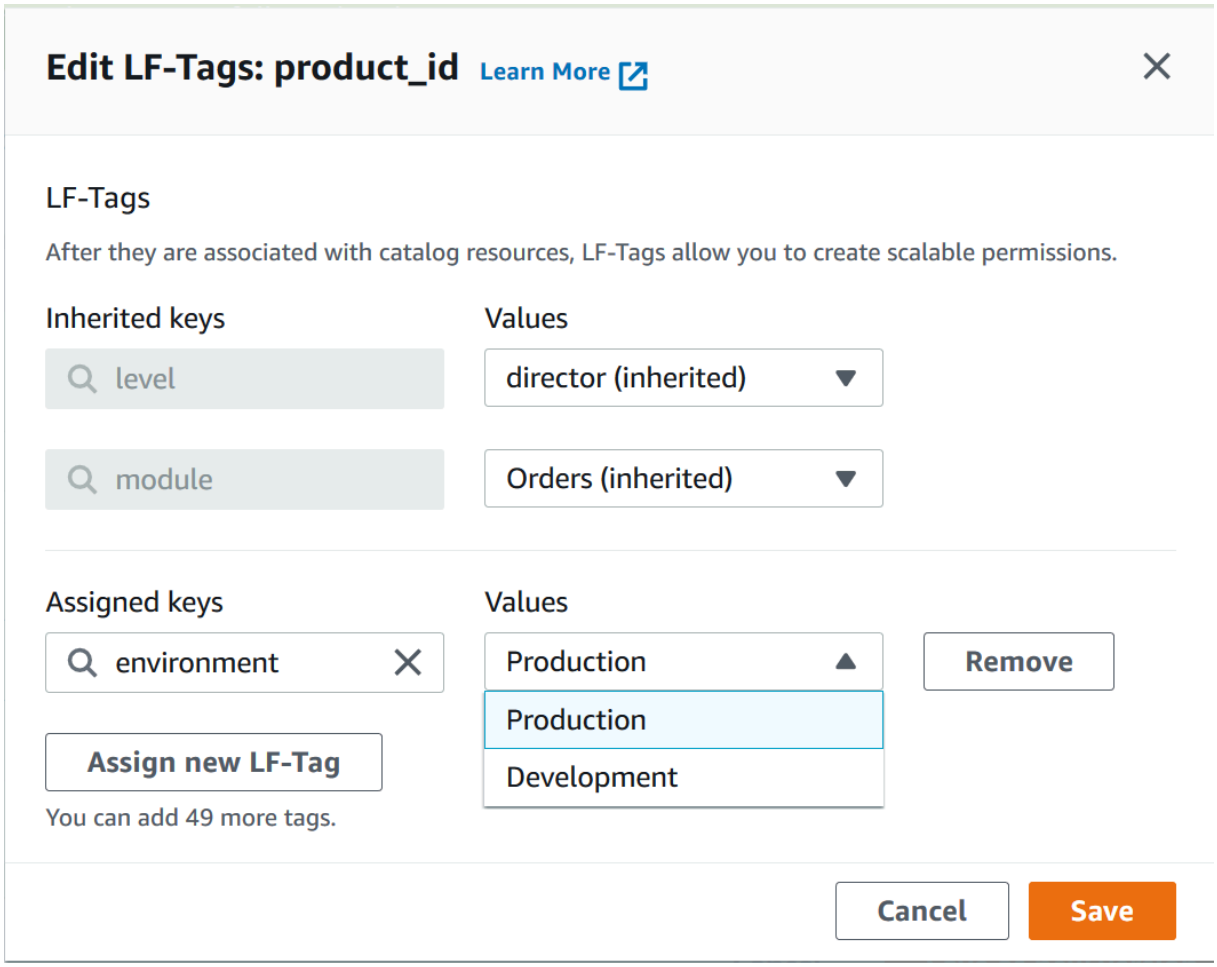
   上記の要件を満たすユーザーとしてサインインします。
2. ナビゲーションペインで [Table] (テーブル) を選択します。
3. テーブル名を選択します (テーブル名の横にあるオプションボタンではありません)。
4. テーブルの詳細ページの [Schema] (スキーマ) セクションで、[Edit schema] (スキーマを編集) を選択します。
5. 「スキーマの編集」 ページで、1 つまたは複数の列を選択してから、[LF タグを編集] を選択します。

   > **Note**
   >
   > 列を追加または削除して、新しいバージョンを保存する予定の場合は、最初にそれらを実行してください。その後、LF タグを編集します。

   [LF タグの編集] ダイアログボックスが表示され、テーブルから継承された LF タグが表示されます。



6. (オプション) [Inherited keys] (継承されたキー) フィールドの横にある[Values](値) リストで、継承された値を上書きする値を選択します。
7. (オプション) [Assign new LF-Tag] (新しい LF タグを割り当てる) を選択します。その後、[Assigned keys] (割り当てられたキー) でキーを選択し、[Values] (値) でキーの値を選択します。



8. (オプション) [新しい LF タグを割り当てる] を再度選択して、別の LF タグを追加します。
9. [Save] (保存) を選択します。

LF タグをデータカタログリソースに割り当てる

Console

LF タグをデータカタログデータベースまたはテーブルに割り当てるには

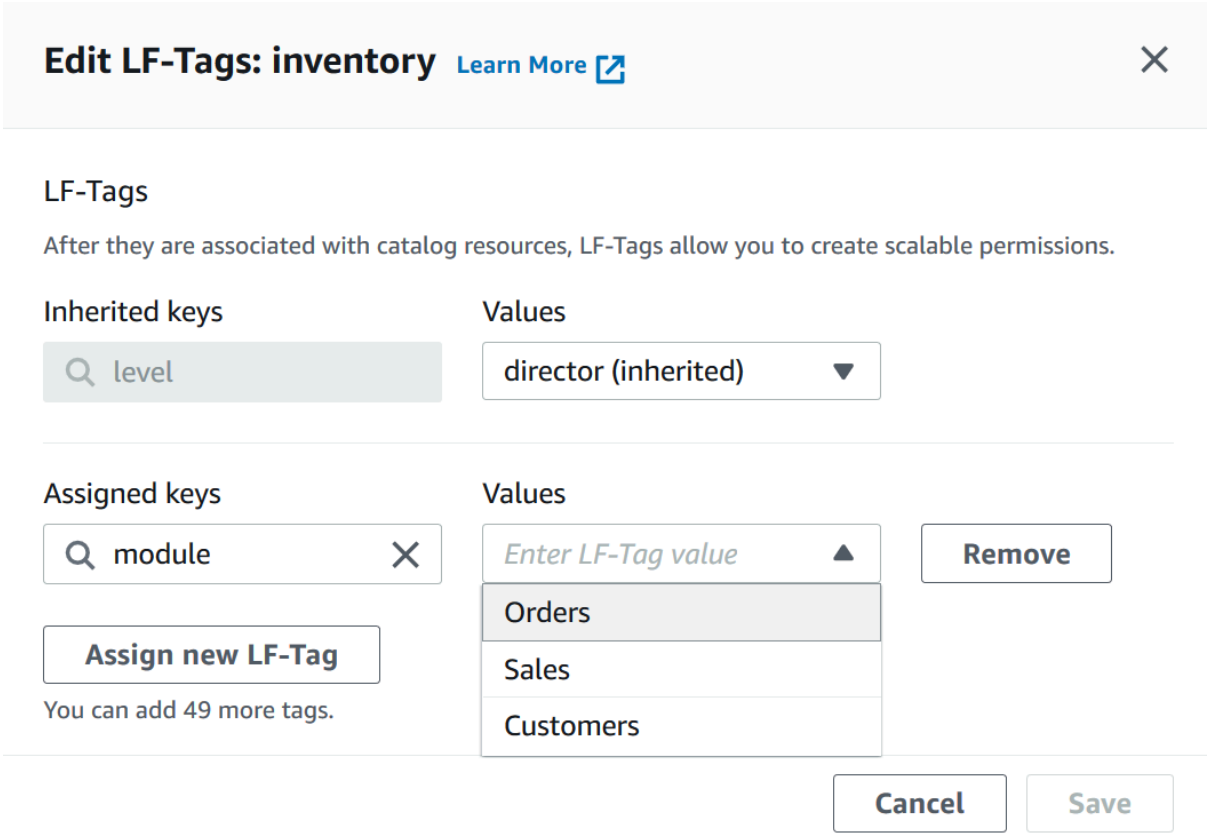
1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   前述の要件を満たすユーザーとしてサインインします。
2. ナビゲーションペインの [Data catalog] で、以下のいずれかを実行します。
   - LF タグをデータベースに割り当てるには、[データベース] を選択します。
   - LF タグをテーブルに割り当てるには、[テーブル] を選択します。

3. データベースまたはテーブルを選択し、[アクション] メニューで [LF タグを編集] を選択します。

   [Edit LF-Tags: ***resource-name***] (LF タグの編集: リソース名) ダイアログボックスが表示されます。

   テーブルが格納先のデータベースから LF タグを継承した場合、継承された LF タグがウィンドウに表示されます。それ以外の場合は、「There are no inherited LF-Tags associated with the resource.」(このリソースに、継承された LF タグは関連付けられていません。) というテキストが表示されます。



4. (オプション) テーブルに継承された LF タグがある場合、[継承されたキー] フィールドの横の [値] リストで、継承された値をオーバーライドする値を選択することができます。
5. 新しい LF タグを割り当てるには、以下の手順を実行します。

   a. [Assign new LF-Tag] (新しい LF タグを割り当てる) を選択します。

   b. [割り当てられたキー] フィールドで LF タグのキーを選択し、[値] フィールドで値を選択します。

c. (オプション) [新しい LF タグを割り当てる] を再度選択して、追加の LF タグを割り当てます。

6. [Save] (保存) を選択します。

AWS CLI

LF タグをデータカタログリソースに割り当てるには

- add-lf-tags-to-resource コマンドを実行します。

次の例は、LF タグ module=orders をデータベース erp 内のテーブル orders に割り当てます。これは、--lf-tags 引数にショートカット構文を使用しています。--lf-tags の CatalogID プロパティはオプションです。指定されない場合は、リソース (この場合はテーブル) のカタログ ID が使用されます。

```
aws lakeformation add-lf-tags-to-resource --resource '{ "Table":
 {"DatabaseName":"erp", "Name":"orders"}}' --lf-tags
 CatalogId=111122223333,TagKey=module,TagValues=orders
```

以下は、コマンドが成功した場合の出力です。

```
{
    "Failures": []
}
```

次の例は、2 つの LF タグを sales テーブルに割り当てて、--lf-tags 引数に JSON 構文を使用します。

```
aws lakeformation add-lf-tags-to-resource --resource '{ "Table":
 {"DatabaseName":"erp", "Name":"sales"}}' --lf-tags '[{"TagKey":
 "module","TagValues": ["sales"]},{"TagKey": "environment","TagValues":
 ["development"]}]'
```

次の例は、LF タグ level=director をテーブル sales の total 列に割り当てます。

```
aws lakeformation add-lf-tags-to-resource --resource '{ "TableWithColumns":
 {"DatabaseName":"erp", "Name":"sales", "ColumnNames":["total"]}}' --lf-tags
 TagKey=level,TagValues=director
```

リソースの LF タグ の更新

データカタログリソースの LF タグを更新するには (AWS CLI)

- 前の手順で説明したように、`add-lf-tags-to-resource` コマンドを使用します。

  既存の LF タグと同じキーを持つが、値は異なるという LF タグを追加すると、既存の値が更新されます。

リソースからの LF タグの削除

データカタログリソースの LF タグを削除するには (AWS CLI)

- `remove-lf-tags-from-resource` コマンドを実行します。

  親データベースから継承された値をオーバーライドする LF タグの値がテーブルにある場合、テーブルからその LF タグを削除すると、継承された値が復元されます。この動作は、テーブルから継承されたキーの値を上書きする列にも該当します。

  以下の例では、`sales` のテーブルの `total` の列から LF タグ `level=director` を削除します。`--lf-tags` の `CatalogID` プロパティはオプションです。指定されない場合は、リソース (この場合はテーブル) のカタログ ID が使用されます。

```
aws lakeformation remove-lf-tags-from-resource
--resource ' { "TableWithColumns":
{ "DatabaseName": "erp",  "Name": "sales",  "ColumnNames":[ "total"]}}'
--lf-tags  CatalogId=111122223333,TagKey=level,TagValues=director
```

リソースに割り当てられた LF タグの表示

データカタログリソースに割り当てられた LF タグを表示できます。LF タグを表示するには、それに対する `DESCRIBE` または `ASSOCIATE` アクセス許可が必要です。

Console

リソースに割り当てられた LF タグを表示するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

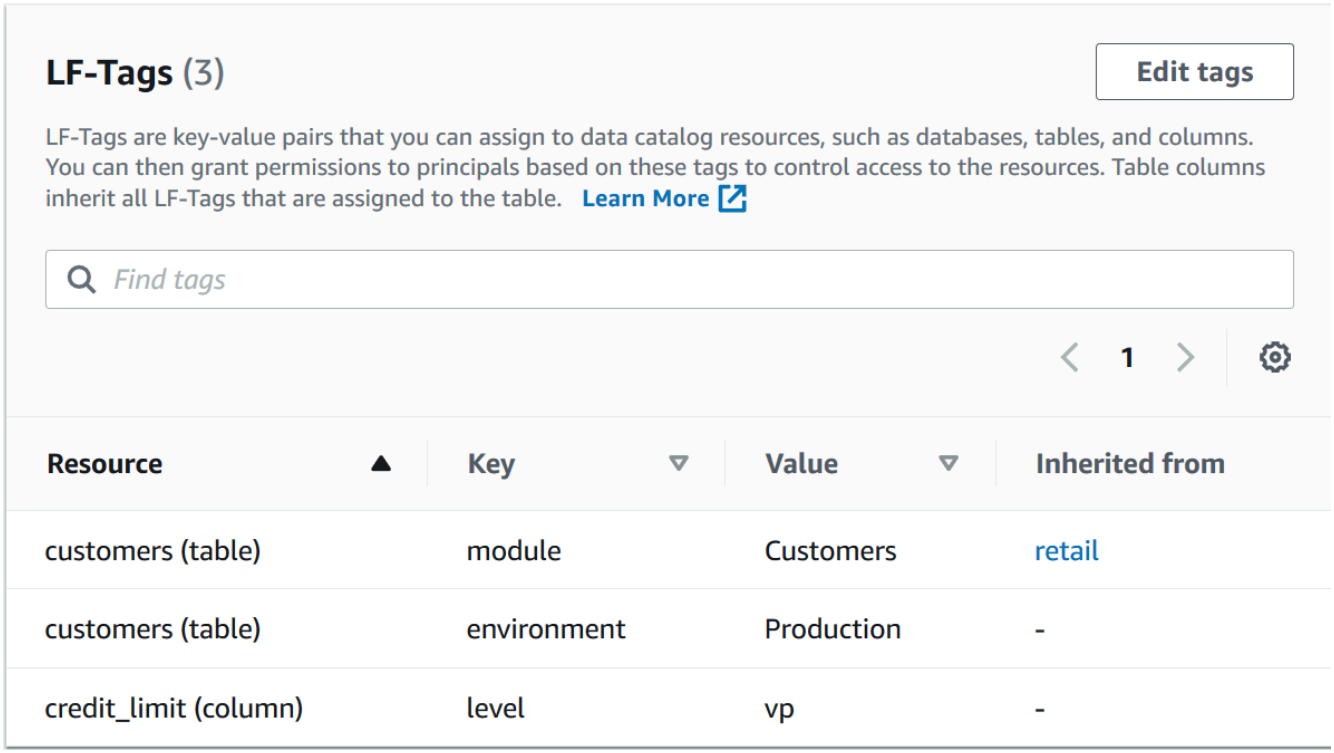
データレイク管理者、リソース所有者、またはリソースに対する Lake Formation 許可を付与されたユーザーとしてサインインします。

2. ナビゲーションペインの [Data catalog] (データカタログ) 見出しの下で、以下のいずれかを実行します。

   - データベースに割り当てられた LF タグを表示するには、[データベース] を選択します。
   - テーブルに割り当てられた LF タグを表示するには、[テーブル] を選択します。

3. [Tables] (テーブル) または [Databases] (データベース) ページで、データベースまたはテーブルの名前を選択します。次に、詳細ページで、[LF-Tags] セクションまでスクロールダウンします。

   次のスクリーンショットは、`retail` データベースに含まれる `customers` テーブルに割り当てられた LF タグを示しています。`module` LF タグはデータベースから継承されます。`credit_limit` 列には `level=vp` LF タグが割り当てられています。



AWS CLI

リソースに割り当てられた LF タグを表示するには (AWS CLI)

- 以下のようなコマンドを入力します。

```
aws lakeformation get-resource-lf-tags --show-assigned-lf-tags --
resource '{ "Table": {"CatalogId":"111122223333", "DatabaseName":"erp",
 "Name":"sales"}}'
```

このコマンドは、以下の出力を返します。

```
{
    "TableTags": [
        {
            "CatalogId": "111122223333",
            "TagKey": "module",
            "TagValues": [
                "sales"
            ]
        },
        {
            "CatalogId": "111122223333",
            "TagKey": "environment",
            "TagValues": [
                "development"
            ]
        }
    ],
    "ColumnTags": [
        {
            "Name": "total",
            "Tags": [
                {
                    "CatalogId": "111122223333",
                    "TagKey": "level",
                    "TagValues": [
                        "director"
                    ]
                }
            ]
        }
    ]
}
```

この出力には、明示的に割り当てられた LF タグのみが表示され、継承されたものは表示されません。継承された LF タグを含め、すべての列のすべての LF タグを表示するには、`--show-assigned-lf-tags` オプションを削除します。

LF タグが割り当てられているリソースの表示

特定の LF タグのキーが割り当てられているすべてのデータカタログリソースを表示できます。これを実行するには、以下の Lake Formation 許可が必要です。

- LF タグに対する `Describe` または `Associate`。
- リソースに対する `Describe`、またはその他 Lake Formation 許可。

さらに、次の AWS Identity and Access Management (IAM) アクセス許可が必要です。

- `lakeformation:SearchDatabasesByLFTags`
- `lakeformation:SearchTablesByLFTags`

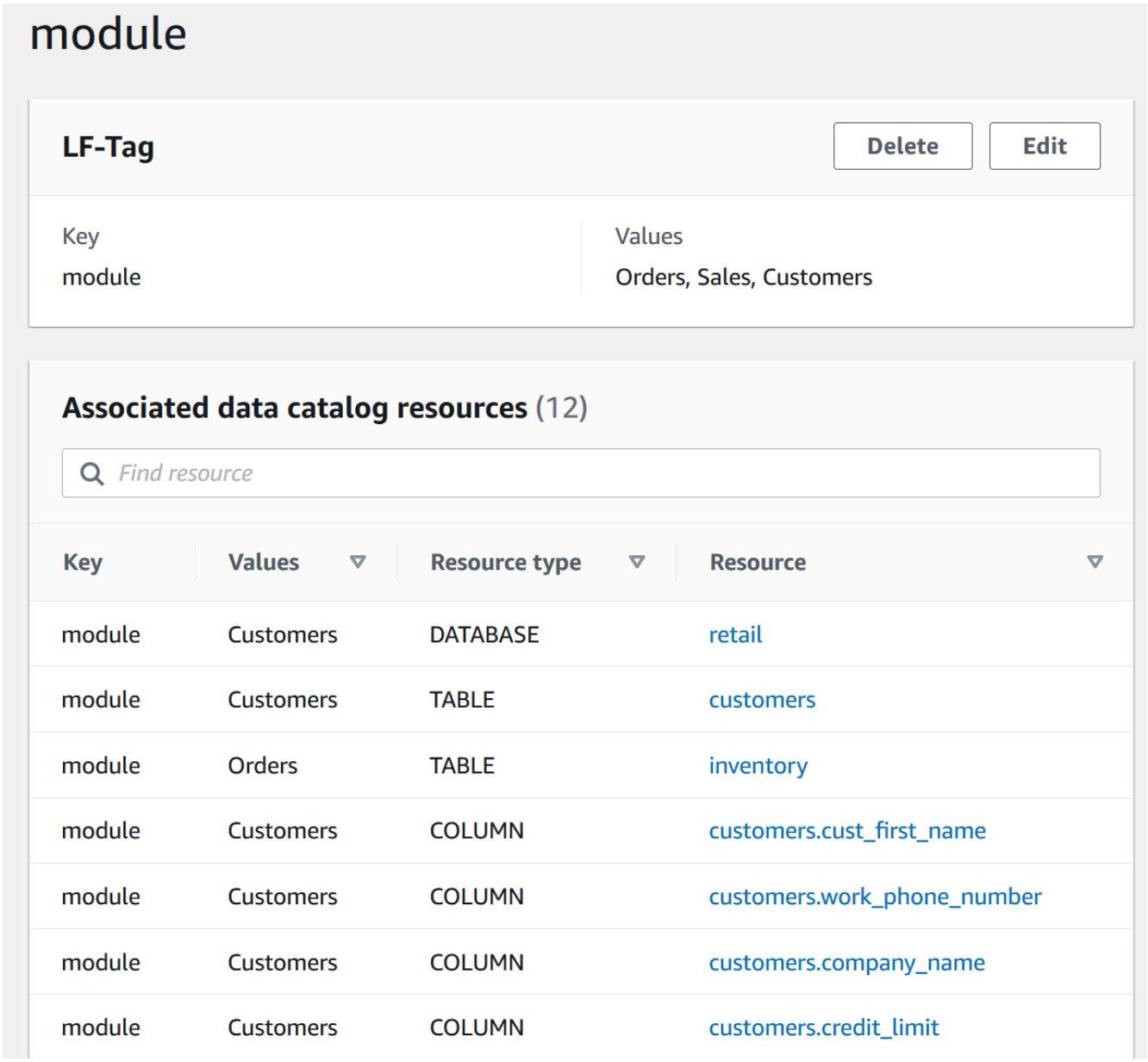
Console

LF タグが割り当てられているリソースを表示するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   データレイク管理者として、または前述の要件を満たすユーザーとしてサインインします。
2. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] の [LF タグ] を選択します。
3. LF タグのキーを選択します (キー名の横にあるオプションボタンではありません)。

   LF タグの詳細ページに、LF タグが割り当てられているリソースのリストが表示されます。



AWS CLI

LF タグが割り当てられているリソースを表示するには

- `search-tables-by-lf-tags` または `search-databases-by-lf-tags` のコマンドを実行します。

Example

次の例は、level=vp LF タグが割り当てられたテーブルと列をリストします。リストされた各テーブルと列について、検索式だけでなく、テーブルまたは列に割り当てられたすべての LF タグが出力されます。

```
aws lakeformation search-tables-by-lf-tags --expression
 TagKey=level,TagValues=vp
```

必要な許可の詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

LF タグのライフサイクル

1. LFタグ作成者のマイケルが LF タグ module=Customers を作成します。
2. マイケルは LF タグに対する Associate をデータエンジニアであるエデュアルドに付与します。Associate の付与は、Describe を黙示的に付与します。
3. マイケルはテーブル Custs に対する grant オプション付きの Super をエデュアルドに付与して、エデュアルドがそのテーブルに LF タグを割り当てることができるようにします。詳細については、「Data Catalog リソースへの LF タグの割り当て」を参照してください。
4. エデュアルドは LF タグ module=customers をテーブル Custs に割り当てます。
5. マイケルがデータエンジニアであるサンドラに以下の付与を行います (疑似コードを使用)。

```
GRANT (SELECT, INSERT ON TABLES) ON TAGS module=customers TO Sandra WITH GRANT OPTION
```

6. サンドラがデータアナリストであるマリアに以下の付与を行います。

```
GRANT (SELECT ON TABLES) ON TAGS module=customers TO Maria
```

マリアが Custs テーブルにクエリを実行できるようになります。

関連情報

- メタデータのアクセスコントロール

Lake Formation のタグベースのアクセス制御と IAM の属性ベースのアクセス制御の比較

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。IAM エンティティ (ユーザーまたはロール) を含む IAM リソースと AWS リソースにタグをアタッチできます。IAM プリンシパルに対して、単一の ABAC ポリシー、または少数のポリシーのセットを作成できます。これらの ABAC ポリシーは、プリンシパルのタグがリソースタグと一致するときに操作を許可するように設計することができます。ATBAC は、急成長する環境や、ポリシー管理が煩雑になる状況で役に立ちます。

クラウドセキュリティチームとガバナンスチームは、Amazon S3 バケット、Amazon EC2 インスタンス、および ARN で参照できるすべてのリソースを含めたすべてのリソースに対するアクセスポリシーとセキュリティ許可を定義するために IAM を使用します。IAM ポリシーは、例えば Amazon S3 バケット、プレフィックスレベル、またはデータベースレベルでアクセスを許可または拒否するために、データレイクリソースに対する広範な (粗粒度の) 許可を定義します。IAM ABAC の詳細については、「ABAC とは」を参照してください AWS。 「IAM ユーザーガイド」の「」を参照してください。

例えば、`project-access` タグキーを使用して 3 つのロールを作成できます。最初のロールのタグ値を `Dev`、2 番目を `Marketing`、3 番目を `Support` に設定します。適切な値を持つタグをリソースに割り当てます。そうすることで、`project-access` に関してロールとリソースが同じ値でタグ付けされているときにアクセスを許可する単一のポリシーを使用できます。

データガバナンスチームは、特定のデータレイクリソースに対するきめ細かな許可を定義するために Lake Formation を使用します。LF タグはデータカタログリソース (データベース、テーブル、および列) に割り当てられ、プリンシパルに付与されます。リソースの LF タグと一致する LF タグを持つプリンシパルは、そのリソースにアクセスできます。Lake Formation 許可は IAM 許可に次ぐ二次的なものです。例えば、IAM アクセス許可がユーザーにデータレイクへのアクセスを許可していない場合、プリンシパルとリソースの LF タグが一致する場合でも、Lake Formation はそのデータレイク内のリソースへのアクセスをそのユーザーに付与しません。

Lake Formation のタグベースのアクセス制御 (LF-TBAC) は、IAM ABAC と連動して Lake Formation のデータとリソースに追加の許可レベルを提供します。

- Lake Formation TBAC 許可は、イノベーションとともにスケールします。新しいリソースへのアクセスを許可するために、管理者が既存のポリシーを更新する必要はありません。例えば、Lake Formation 内で特定のデータベースへのアクセスを提供するために `project-access` タグでの IAM ABAC 戦略を使用するとします。LF-TBAC を使用することで、LF タグ`Project=SuperApp` は特定のテーブルや列に 割り当てられ、同じ LF タグがそのプロジェクトのデベロッパーに付与されます。デベロッパーは IAM を通じてデータベースにアクセスでき、LF-

TBAC 許可はこのデベロッパーに特定のテーブル、またはテーブル内の列への追加のアクセス権を付与します。新しいテーブルがプロジェクトに追加される場合、Lake Formation 管理者は、新しいテーブルにタグを割り当てて、デベロッパーにそのテーブルへのアクセス権が付与されるようするだけで済みます。

- Lake Formation TBAC では、必要な IAM ポリシーが少なくなります。ユーザーは Lake Formation リソースに対する高レベルのアクセス権を付与するために IAM ポリシーを使用し、より精密なデータアクセスの管理のために Lake Formation TBAC を使用するので、作成する IAM ポリシーが少なくなります。
- Lake Formation TBAC を使用することで、チームのより迅速な変化と成長が可能になります。これは、新しいリソースの許可が属性に基づいて自動的に付与されるためです。例えば、新しいデベロッパーがプロジェクトに参加する場合、IAM ロールをユーザーに関連付けてから、必要な LF タグをユーザーに割り当てることで、このデベロッパーにアクセス権を簡単に付与できます。新しいプロジェクトをサポートするためや、新しい LF タグを作成するために IAM ポリシーを変更する必要はありません。
- Lake Formation TBAC を使用することで、よりきめ細かな許可が可能になります。IAM ポリシーは、Data Catalog のデータベースやテーブルなどのトップレベルリソースへのアクセス権を付与します。Lake Formation TBAC を使用することで、特定のデータ値が含まれる特定のテーブルやカラムにアクセス権を付与することができます。

> **Note**
>
> IAM タグと LF タグは同じではありません。これらのタグは置き換え可能ではありません。LF タグは Lake Formation アクセス許可を付与するために使用され、IAM タグは IAM ポリシーを定義するために使用されます。

メタデータアクセスコントロールのための LF タグ式の管理

LF タグ式は、 AWS Glue Data Catalog リソースに対するアクセス許可を付与するために使用される 1 つ以上の LF タグ (キーと値のペア) で構成される論理式です。LF タグ式を使用すると、メタデータタグに基づいてデータリソースへのアクセスを管理するルールを定義できます。これらの式を保存して複数のアクセス許可で再利用できるため、一貫性が確保され、タグオントロジーへの経時的な変更を簡単に管理できます。

特定の LF タグ式内では、タグキーは AND オペレーションを使用して結合され、値は OR オペレーションを使用して結合されます。例えば、タグ式 `content_type:Sales AND location:US` は、米国の売上データに関連するリソースを表します。

AWS アカウントでは、最大 1000 個の LF タグ式を作成できます。これらの式は、メタデータタグに基づいてアクセス許可を管理するための柔軟でスケーラブルな方法を提供し、承認されたユーザーまたはアプリケーションのみが、定義されたタグルールに基づいて特定のデータリソースにアクセスできるようにします。

LF タグ式には以下のメリットがあります。

- 再利用性 – LF タグ式を定義して保存することで、他のリソースやプリンシパルにアクセス許可を割り当てるときに、同じ式を手動でレプリケートする必要がなくなります。
- 整合性 – 複数のアクセス許可の付与で LF タグ式を再利用することで、アクセス許可の付与と管理の一貫性を確保できます。
- タグオントロジー管理 – LF タグ式は、個々のアクセス許可付与を変更する代わりに保存された式を更新できるため、時間の経過に伴うタグオントロジーへの変更を管理するのに役立ちます。

タグベースのアクセスコントロールに関する詳細については、「Lake Formation のタグベースのアクセス制御」を参照してください。

LF タグ式作成者

LF タグ式作成者は、LF タグを作成および管理するアクセス許可を持つプリンシパルです。データレイク管理者は、Lake Formation コンソール、CLI、API、または SDK を使用して LF タグ式作成者を追加できます。LF タグ式作成者には、LF タグ式を作成、更新、削除し、LF タグ式アクセス許可を他のプリンシパルに付与する暗黙的な Lake Formation アクセス許可があります。

データレイク管理者ではない LF タグ式作成者は、作成した式に対してのみ暗黙的な `Alter`、`Drop`、`Describe`、および `Grant with LF-Tag expression` アクセス許可を受け取ります。

データレイク管理者は LF タグ式作成者に付与可能な `Create LF-Tag expression` アクセス許可を付与することもできます。その後、LF タグ式作成者は、LF タグ式を作成するアクセス許可を他のプリンシパルに付与できます。

トピック

- LF タグ式の作成に必要な IAM アクセス許可

- LF タグ式作成者を追加する
- LF タグ式の作成
- LF タグ式の更新
- LF タグ式の削除
- LF タグ式の一覧表示

> **関連情報**
>
> - LF タグ値のアクセス許可の管理
> - LF-TBAC 方式を使用したデータレイク許可の付与
> - Lake Formation のタグベースのアクセス制御

LF タグ式の作成に必要な IAM アクセス許可

Lake Formation のプリンシパルが LF タグ式を作成できるようにアクセス許可を設定する必要があります。LF タグ式作成者になる必要があるプリンシパルのアクセス許可ポリシーに、以下のステートメントを追加します。

> **Note**
>
> データレイク管理者は、LF タグと LF タグ式の作成、更新、削除、LF タグのリソースへの割り当て、プリンシパルへの LF タグと LF タグ式の付与を行う暗黙的な Lake Formation アクセス許可を持っていますが、データレイク管理者には以下の IAM アクセス許可も必要です。

詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

```
{
"Sid": "Transformational",
"Effect": "Allow",
    "Action": [
        "lakeformation:AddLFTagsToResource",
        "lakeformation:RemoveLFTagsFromResource",
        "lakeformation:GetResourceLFTags",
```

```
        "lakeformation:ListLFTags",
        "lakeformation:CreateLFTag",
        "lakeformation:GetLFTag",
        "lakeformation:UpdateLFTag",
        "lakeformation:DeleteLFTag",
        "lakeformation:SearchTablesByLFTags",
        "lakeformation:SearchDatabasesByLFTags",
        "lakeformation:CreateLFTagExpression",
        "lakeformation:DeleteLFTagExpression",
        "lakeformation:UpdateLFTagExpression",
        "lakeformation:GetLFTagExpression",
        "lakeformation:ListLFTagExpressions",
        "lakeformation:GrantPermissions",
        "lakeformation:RevokePermissions",
        "lakeformation:BatchGrantPermissions",
        "lakeformation:BatchRevokePermissions"
    ]
}
```

LF タグ式作成者を追加する

LF タグ式作成者は、再利用可能な LF タグ式の作成と保存、タグのキーと値の更新、式の削除、LF-TBAC 方式を使用したプリンシパルへのデータカタログリソースに対するアクセス許可の付与を行うことができます。LF タグ式作成者は、これらのアクセス許可をプリンシパルに付与することもできます。

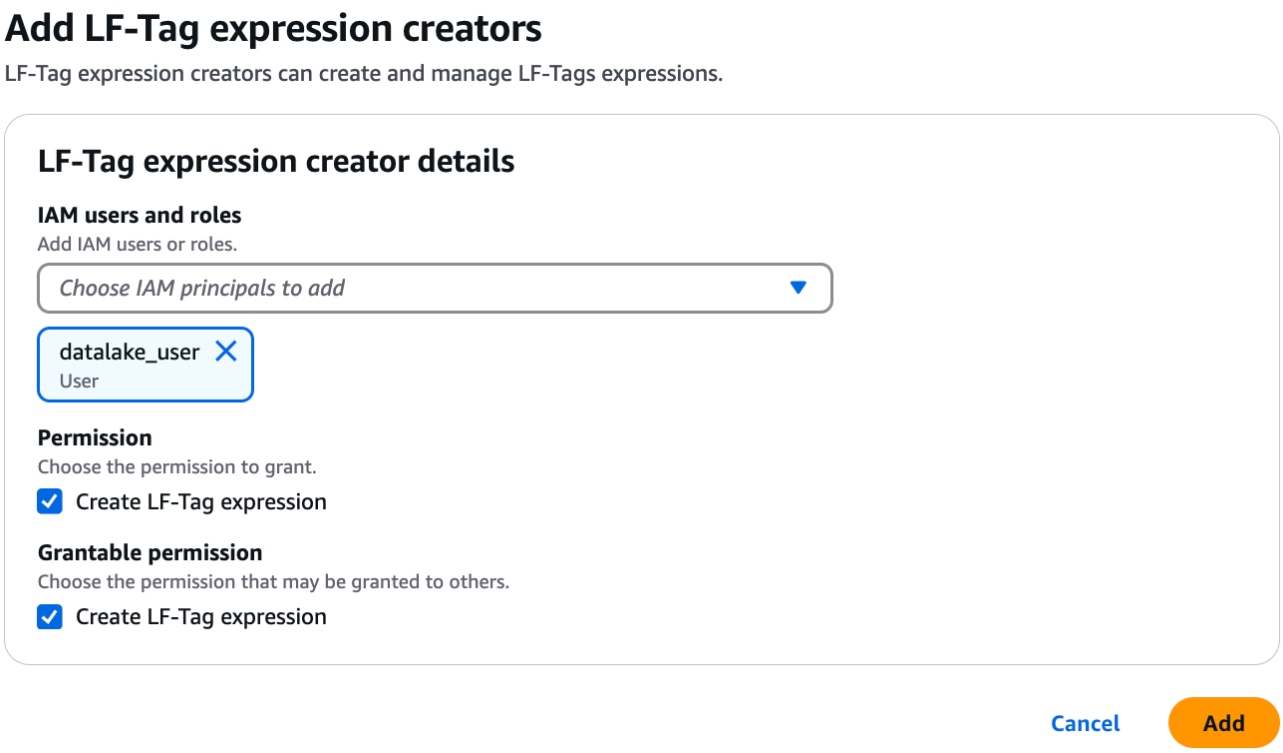
AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、LF タグ式作成者ロールを作成できますAWS CLI。

console

LF タグ式作成者を追加するには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   データレイク管理者としてサインインします。
2. ナビゲーションペインで、[アクセス許可] の [LF タグとアクセス許可] を選択します。
3. [LF タグ式] タブを選択します。
4. [LF タグ式作成者]セクションで、[LF タグ式作成者を追加] を選択します。



5. 「LF タグ式作成者を追加」ページで、LF タグ式の作成に必要なアクセス許可を持つ IAM ロールまたはユーザーを選択します。
6. [Create LF-Tag expression アクセス許可] チェックボックスにチェックを入れます。
7. (オプション) 選択したプリンシパルが Create LF-Tag expression アクセス許可をプリンシパルに付与できるようにするには、[付与可能な Create LF-Tag expression アクセス許可] を選択します。
8. [Add] (追加) を選択します。

AWS CLI

```
aws lakeformation grant-permissions --cli-input-json file://grantCreate
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789012:user/tag-manager"
    },
    "Resource": {
        "Catalog": {}
    },
    "Permissions": [
        "CreateLFTagExpression"
```

```
    ],
    "PermissionsWithGrantOption": [
        "CreateLFTagExpression"
    ]
}
```

LF タグ式作成者ロールは、LF タグ式を作成、更新、または削除することができます。

アクセス許可	説明
Create	このアクセス許可を持つプリンシパルは、データレイクに LF タグ式を追加できます。
Drop	LF タグ式に対するこのアクセス許可を持つプリンシパルは、データレイクから LF タグ式を削除できます。
Alter	LF タグ式に対するこのアクセス許可を持つプリンシパルは、LF タグ式の式本文を更新できます。
Describe	LF タグ式に対するこのアクセス許可を持つプリンシパルは、LF タグ式の内容を表示できます。
Grant with LF-Tag expression	このアクセス許可により、受信者はデータまたはメタデータにアクセス許可を付与するときに、タグ式をリソースとして使用できます。Grant with LF-Tag expression の付与は、Describe を黙示的に付与します。
Super	LF タグ式の場合、Super ではタグ式に対する Describe、Alter、Drop の操作、および他のプリンシパルへのアクセス許可の付与が可能になります。

これらの許可は付与可能です。これらの許可を grant オプションと共に付与されたプリンシパルは、これらを他のプリンシパルに付与できます。

LF タグ式の作成

Lake Formation ですべての LF タグを定義し、式の作成に使用する前にデータカタログリソースに割り当てる必要があります。LF タグ式は、キーごとに 1 つ以上のキーと 1 つ以上の可能な値で構成されます。

データレイク管理者が LF タグ式作成者ロールに必要な IAM アクセス許可と Lake Formation アクセス許可を設定したら、プリンシパルは再利用可能な LF タグ式を作成できます。LF タグ式作成者は、式本文を更新し、LF タグ式を削除するための暗黙的なアクセス許可を取得します。

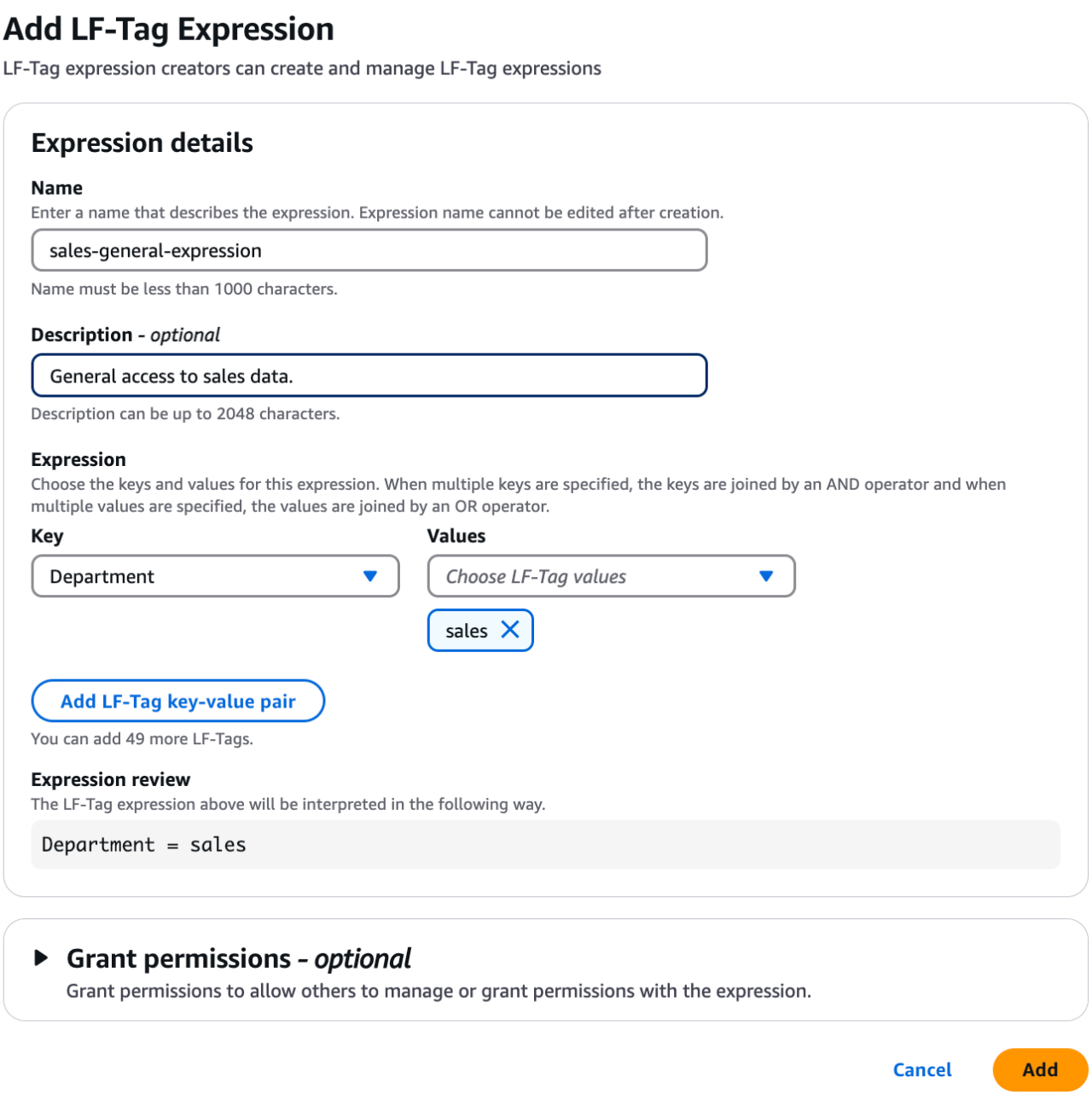
LF タグ式は、 AWS Lake Formation コンソール、 API、または AWS Command Line Interface () を使用して作成できますAWS CLI。

Console

LF タグ式を作成するには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   LF タグ式作成者アクセス許可を持つプリンシパルまたはデータレイク管理者としてサインインします。
2. ナビゲーションペインのアクセス許可で、LF タグとアクセス許可を選択します。
3. [LF タグ式] を選択します。「LF タグ式の追加」ページが表示されます。



4. 次の情報を入力します。

   - 名前 - 式の一意の名前を入力します。式名を更新することはできません。
   - 説明 – 式のオプションの説明と式の詳細を入力します。
   - 式 – タグキーとそれに関連する値を指定して式を作成します。式ごとに最大 50 個のキーを追加できます。式本文内のすべてのタグに対して `Grant with LF-Tags` Lake Formation アクセス許可が必要です。

各キーには、少なくとも 1 つの値が必要です。複数の値を入力するには、カンマ区切りのリストを入力してから [Enter] キーを押すか、一度に 1 つの値を入力し、入力するたびに [Add] (追加) を選択します。キーごとに許可される値の最大数は 1000 個です。

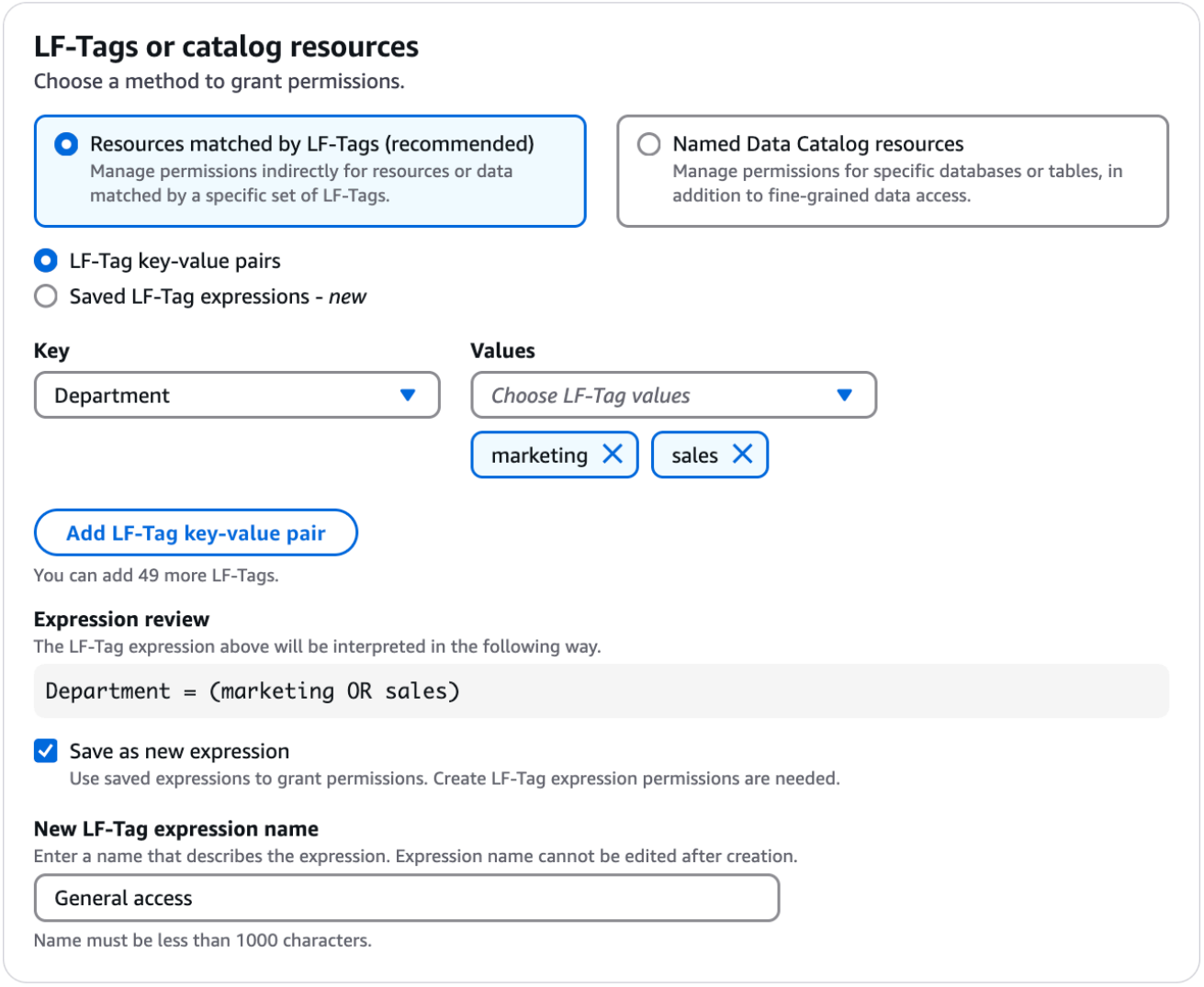
Lake Formation は AND/OR ロジックを使用して、式内の複数のキーと値を結合します。単一の (キー: 値のリスト) ペア内では、論理 OR 演算子を使用して値が結合されます。例えば、ペアが (Department : [Sales, Marketing]) の場合、リソースに Sales OR Marketing の値が指定された Department タグがある場合に、タグが一致します。

複数のキーを指定すると、キーは AND 論理演算子によって結合されます。したがって、完全な式が (Department : [Sales, Marketing]) AND (Location : [US, Canada]) の場合、Sales OR Marketing という値を持つ Department タグと、米国またはカナダという値を持つ Location タグを持つリソースに一致します。以下は複数のキーと値を持つ別の例です。

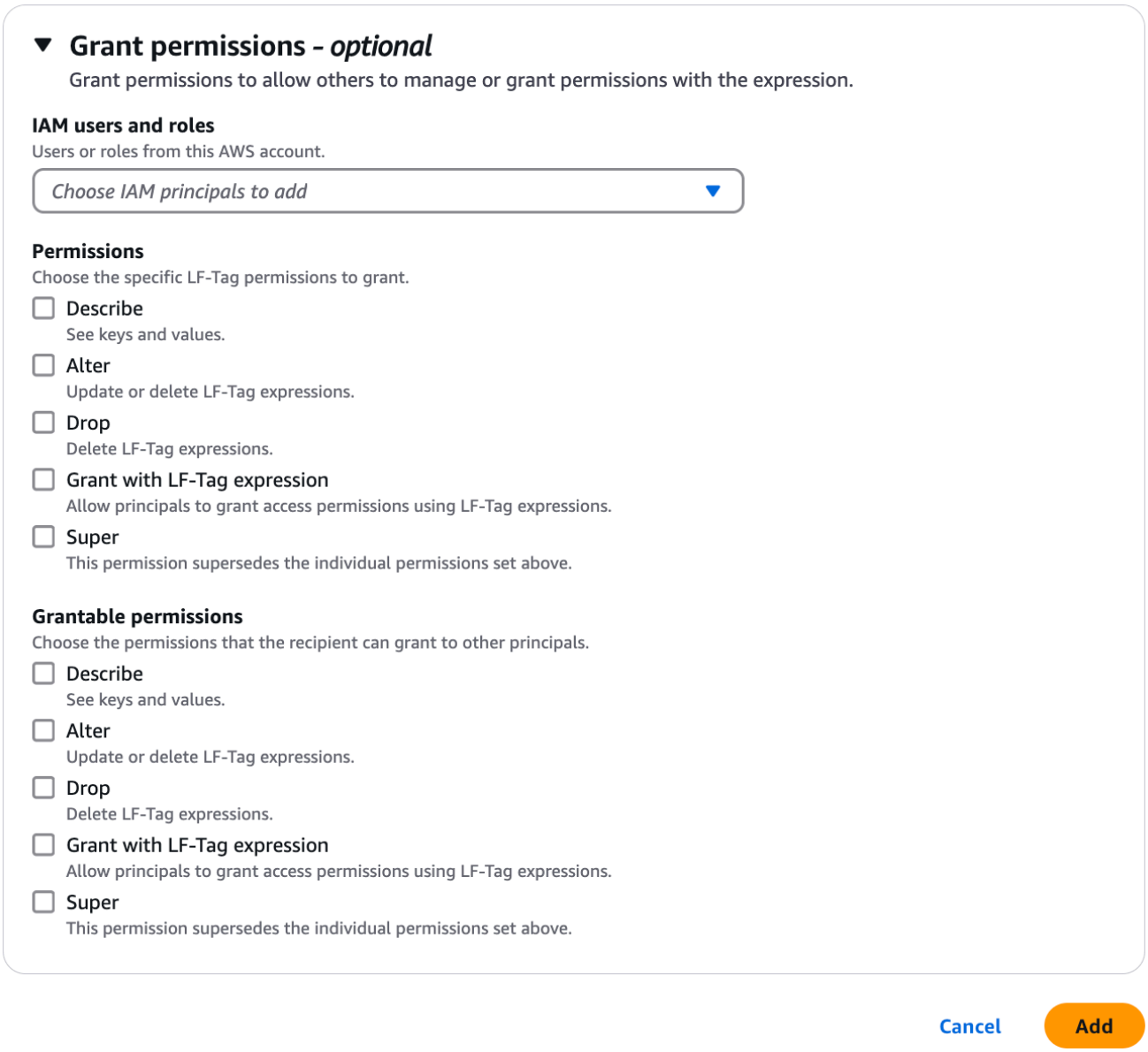
LF タグ式: (ContentType : [Video, Audio]) AND (Region : [Europe, Asia]) AND (Department : [Engineering, ProductManagement])。

この式は、 - 値が Video OR Audio の ContentType タグ AND - 値が Europe OR Asia のリージョンタグ AND - 値が Engineering OR ProductManagement の Department タグを持つリソースと一致します。

LF タグを使用してデータレイクのアクセス許可を付与するときに、タグ式を保存することもできます。キーと値のペアを選択し、[新しい式として保存] オプションを選択します。式を説明する名前を入力します。



5. (オプション) 次に、ユーザー/ロールと、アカウントでユーザー/ロールに付与する式のアクセス許可を選択します。また、付与可能なアクセス許可を選択して、ユーザーがアカウントの他のユーザーにこれらのアクセス許可を付与できるようにします。タグ式にクロスアカウントアクセス許可を付与することはできません。



6. [の追加] を選択します。

AWS CLI

LF タグ式を作成するには

- `create-lf-tag-expression` のコマンドを入力します。

  次の例では、値 `Sales` と `Marketing` を持つ `Department` タグと、値 `US` を持つタグ `Location` を使用して LF タグ式を作成します。

```
aws lakeformation create-lf-tag-expression \
```

```
-- name "my-tag-expression" \
-- catalog-id "123456789012" \
-- expression '{"Expression":[{"TagKey":"Department","TagValues":
["Sales","Marketing"]},{"TagKey":"Location","TagValues":["US"]}]}'
```

この CLI コマンドは、 に新しい LF タグ式を作成します AWS Glue Data Catalog。式を使用すると、データベース、テーブル、ビュー、列などのデータカタログリソースに、関連するタグに基づいてアクセス許可を付与できます。この例では、式は、`Department` キー が値 `Sales` または `Marketing` で、`Location` キー が値 `US` であるリソースと一致します。

プリンシパルは、タグ式作成者 としてこの LF タグ式に対する `Alter` アクセス許可を取得し、式を更新または削除できます。LF タグ作成者プリンシパルは、この式を更新および削除する `Alter` アクセス許可を他のプリンシパルに付与することもできます。

LF タグ式の更新

データレイク管理者と、LF タグ式作成者、LF タグ式に対する `Alter` または `Super` のアクセス許可を持つプリンシパルのみが LF タグ式を更新できます。式を更新するには、`Alter` アクセス許可に加えて、新しい式本文の基礎となるすべてのキーと値に対する `lakeformation:UpdateLFTagExpression` IAM アクセス許可と `Grant with LF-Tag` アクセス許可も必要です。

LF タグ式を更新するには、式で付与された説明、式本文、アクセス許可を更新します。LF タグ式の名前を変更することはできません。名前を変更するには、LF タグ式を削除して、必要なパラメータを持つ LF タグ式を追加します。

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して LF タグ式を更新できますAWS CLI。

Console

LF タグ式を更新するには

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、LF タグ作成者、または LF タグに対する `Alter` アクセス許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインで、[アクセス許可] の[LF タグとアクセス許可] を選択します。
3. [LF タグ式] タブを選択します。

4. [LF タグ式] セクションで、LF タグ式を選択し、次に [編集]を選択します。
5. [LF タグ式を編集] ダイアログボックスで、説明を更新し、キーと値を追加または削除して式本文を更新します。

   複数の値を追加するには、[値] フィールドで、ドロップダウンから値を選択します。
6. [保存] を選択します。

AWS CLI

Lake Formation の update-lf-tag-expression コマンドを使用すると、既存の LF タグ式を更新できます。

```
aws lakeformation update-lf-tag-expression \
-- name expression_name\
-- description new_description \
-- catalog-id catalog_id \
-- expression '{"Expression": [{"TagKey": "tag_key", "TagValues": ["tag_value1",
 "tag_value2", ...]}]}'
```

提供されたコマンドのパラメータの意味は次のとおりです。

- name – 更新する既存の名前付きタグ式の名前。
- description – 式の新しい説明。

  catalog-id – 名前付きタグ式が存在するデータカタログの ID。
- expression – 式を更新する新しいタグ式文字列。

LF タグ式の削除

使用しなくなった LF タグ式は削除できます。LF タグ式を使用してデータカタログリソースのプリンシパルにアクセス許可を付与している場合、プリンシパルのアクセス許可は失われます。

データレイク管理者と、LF タグ式作成者、LF タグ式に対する Drop のアクセス許可を持つプリンシパルのみが LF タグ式を削除できます。プリンシパルが LF タ式グを削除するには、Drop アクセス許可のほかに、lakeformation:DeleteLFTagExpression IAM アクセス許可も必要です。

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、LF タグ式を削除できますAWS CLI。

Console

LF タグ式を削除するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、LF タグ式作成者、または式を削除する権限を持つプリンシパルとしてサインインします。
2. ナビゲーションペインで、[アクセス許可] の [LF タグとアクセス許可] を選択します。
3. [LF タグ式] タブを選択します。
4. [LF タグ式] セクションで、LF タグ式を選択し、[削除] を選択します。
5. [タグ式を削除しますか?] ダイアログボックスで、削除を確定するには、LF タグ式の名前を指定フィールドに入力し、[削除] を選択します。

AWS CLI

LF タグを削除するには (AWS CLI)

- `delete-lf-tag-expression` のコマンドを入力します。削除する式名とカタログ ID を指定します。

  Example

  次の例では、ID 123456789012 の データカタログから `my-tag-expression` という名前の LF タグ式を削除します。 AWS CLI 設定と同じアカウントを使用している場合、`catalog-id`パラメータはオプションです。LF タグ式を削除すると、Lake Formation はその式の関連するアクセス許可レコードをクリーンアップします。これには、削除された式を含む個別のアクセス許可レコードとアクセス許可レコードの集合の両方が含まれます。

  ```
  aws lakeformation delete-lf-tag-expression \
  --name "my-tag-expression" \
  --catalog-id "123456789012"
  ```

LF タグ式の一覧表示

Describe アクセス許可がある LF タグ式を一覧表示することができます。データレイク管理者、LF タグ式作成者、読み取り専用管理者は、自分のアカウント内のすべてのタグ式を暗黙的に表示できます。

AWS Lake Formation コンソール、API、または () を使用して、LF タグ式を AWS Command Line Interface 一覧表示できますAWS CLI。

Console

LF タグ式を一覧表示するには (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   LF タ式グ作成者、データレイク管理者、または LF タグ式に対するアクセス許可を付与され、`lakeformation:ListLFTagExpressions` IAM 許可を持つプリンシパルとしてサインインします。

2. ナビゲーションペインの [アクセス許可] で、[LF タグとアクセス許可] を選択します。
3. [LF タグ式] タブを選択すると、式が表示されます。このセクションには、式名、付けられているタグへのリンクを含む式自体、式を作成、編集、削除するオプションなど、既存の LF タグ式に関する情報が表示されます。

AWS CLI

LF タグをリストする (AWS CLI)

- を使用して LF タグ式を一覧表示するには AWS CLI、list-lf-tag-expressions コマンドを使用できます。リクエスト構文は次のとおりです。

```
aws lakeformation list-lf-tag-expressions \
-- catalog-id "123456789012" \
-- max-items "100" \
-- next-token "next-token"
```

説明は以下のとおりです。

- `catalog-id` は、 のタグ式を一覧表示するデータカタログの AWS アカウント ID です。
- `max-items` は、返されるタグ式の最大数を指定します。このパラメータを使用しない場合、デフォルト値は 100 です。
- `next-token` は、前のリクエストで結果が切り捨てられた場合の継続トークンです。

レスポンスには、LF タグ式のリストと、該当する場合は次のトークンが含まれます。

LF タグ値のアクセス許可の管理

LF タグ値の式を管理するために LF タグに対する Drop、Alter アクセス許可をプリンシパルに付与することができます。プリンシパルが LF タグを表示し、データカタログリソース (データベース、テーブル、列) に割り当てることができるように、LF タグに対する Describe、Associate、および Grant with LF-Tag expressions アクセス許可を付与することもできます。LF タグがデータカタログリソースに割り当てられているときには、Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方法を使用して、これらのリソースをセキュリティで保護できます。詳細については、「Lake Formation のタグベースのアクセス制御」を参照してください。

これらのアクセス許可を grant オプションと共に付与されたプリンシパルは、これらを他のプリンシパルに付与できます。Grant with LF-Tag expressions、Describe、および Associate アクセス許可は、「LF タグ作成者の追加」で説明されています。

LF タグに対する Describeおよび アクセスAssociate許可を外部 AWS アカウントに付与できます。そうすると、そのアカウントのデータレイク管理者が、アカウント内の他のプリンシパルにこれらの許可を付与できるようになります。外部アカウントのデータレイク管理者が Associate アクセス許可を付与したプリンシパルは、アカウントを共有するデータカタログリソースに LF タグを割り当てることができます。

外部アカウントに付与するときは、grant オプションを含める必要があります。

LF タグに対するアクセス許可は、Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して付与することができます。

トピック

- コンソールを使用した LF-Tag アクセス許可の表示
- コンソールを使用した LF-Tag アクセス許可の付与
- を使用した LF タグのアクセス許可の管理 AWS CLI

詳細については、「メタデータアクセスコントロールのための LF タグの管理」および「Lake Formation のタグベースのアクセス制御」を参照してください。

コンソールを使用した LF-Tag アクセス許可の表示

Lake Formation コンソールを使用して、LF タグに付与された許可を表示できます。これを表示するには、LF タグ作成者またはデータレイク管理者であるか、LF タグに対する Describe または Associate アクセス許可を持ってる必要があります。

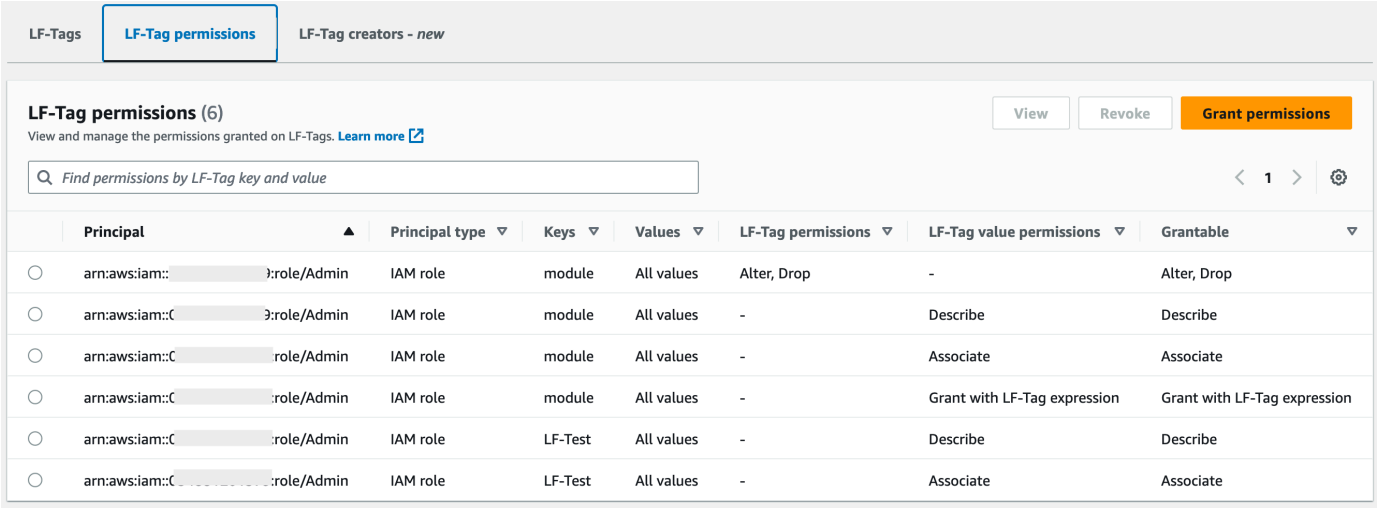
LF タグ許可をリストする (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/)を開きます。

   LF タグ作成者、データレイク管理者、または LF タグに対する `Drop`、`Alter`、`Associate`、または `Describe` 許可が付与されたユーザーとしてサインインします。

2. ナビゲーションペインで、[アクセス許可] の [LF タグとアクセス許可] を選択し、[LF タグアクセス許可] セクションを選択します。

   [LF タグアクセス許可] セクションには、プリンシパル、タグキー、値、およびアクセス許可を含むテーブルが表示されます。



コンソールを使用した LF-Tag アクセス許可の付与

以下の手順では、Lake Formation コンソールの [LF タグアクセス許可の付与] ページを使用して LF タグに対するアクセス許可を付与する方法を説明します。このページは、これらのセクションに分けられています。

- アクセス許可タイプ — 付与するアクセス許可のタイプ。
- プリンシパル – アクセス許可を付与する IAM ユーザーもしくはロール、または SAML ユーザーもしくはロール。
- LF タグキーと値のペアのアクセス許可 – アクセス許可を付与する LF タグキーと値のペア。
- LF タグのアクセス許可 - アクセス許可を付与する LF タグ。
- LF タグ式のアクセス許可 – アクセス許可を付与する LF タグ。
- [Permissions] (許可) – 付与する許可。

[LF タグアクセス許可の付与] ページを開く

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   LF タグ作成者、データレイク管理者、または `Grant` オプションで LF タグアクセス許可または LF タグに対する LF タグのキーと値のペアのアクセス許可が付与されたユーザーとしてサインインします。
2. ナビゲーションペインで、[LF タグとアクセス許可] を選択し、[LF タグアクセス許可] セクションを選択します。
3. [Grant permissions] (アクセス許可の付与) を選択します。

アクセス許可タイプを指定する

[アクセス許可タイプ] セクションで、アクセス許可タイプを選択します。

LF タグアクセス許可

[LF タグアクセス許可] を選択して、プリンシパルが LF タグ値を更新したり、LF タグを削除したりするのを許可します。

LF タグのキーと値のペアのアクセス許可

[LF タグのキーと値のペアのアクセス許可] を選択して、プリンシパルが LF タグをデータカタログリソースに割り当てたり、LF タグと値を表示したり、データカタログリソースに対する LF タグベースのアクセス許可をプリンシパルに付与したりするのを許可します。

以下のセクションで使用できるオプションは、[アクセス許可タイプ] によって異なります。

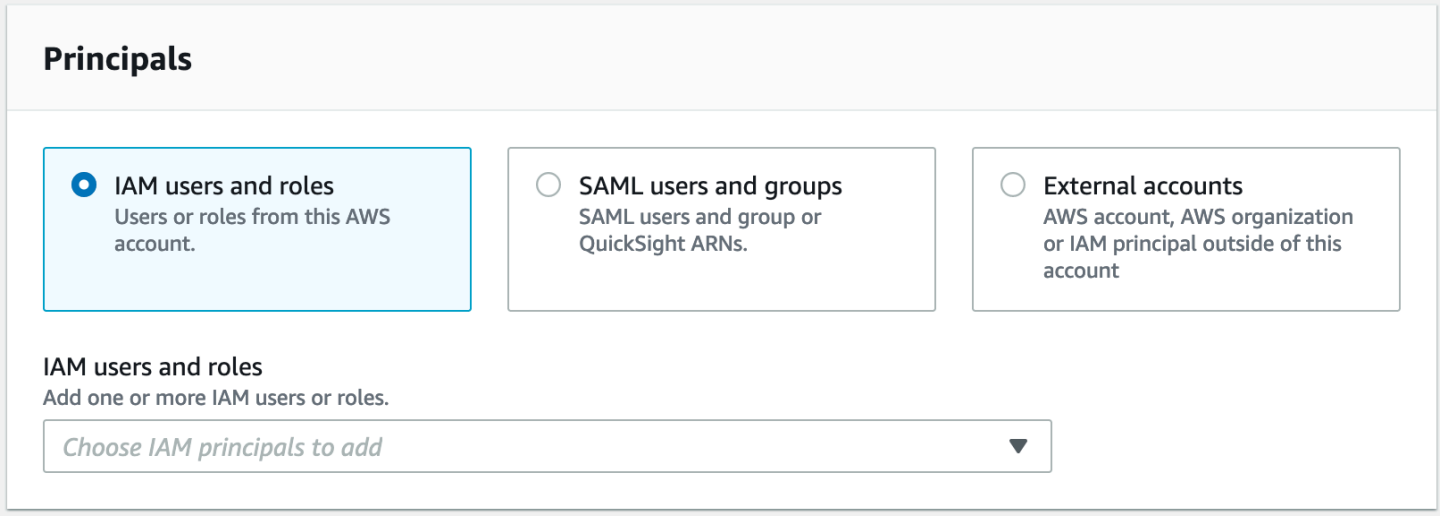
LF タグ式のアクセス許可

[LF タグ式のアクセス許可] を選択して、プリンシパルが式を更新または削除できるようにします。

プリンシパルを指定する

> **Note**
>
> LF タグアクセス許可 (`Alter` および `Drop`) を外部アカウントまたは別のアカウントのプリンシパルに付与することはできません。

[Principals] (プリンシパル) セクションでプリンシパルタイプを選択して、許可の付与先となるプリンシパルを指定します。



IAM ユーザーとロール

[IAM users and roles] (IAM ユーザーおよびロール) リストから、1 人、または複数のユーザーまたはロールを選択します。

SAML ユーザーとグループ

SAML およびクイックユーザーおよびグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、またはクイックユーザーまたはグループの ARNs を入力します。各 ARN の後で [Enter] キーを押します。

ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。

> **Note**
>
> Lake Formation と Quick の統合は、Quick Enterprise Edition でのみサポートされています。

外部アカウント

AWS アカウントには、1 つ以上の有効な AWS アカウント IDsを入力します。各 ID の後で [Enter] キーを押します。

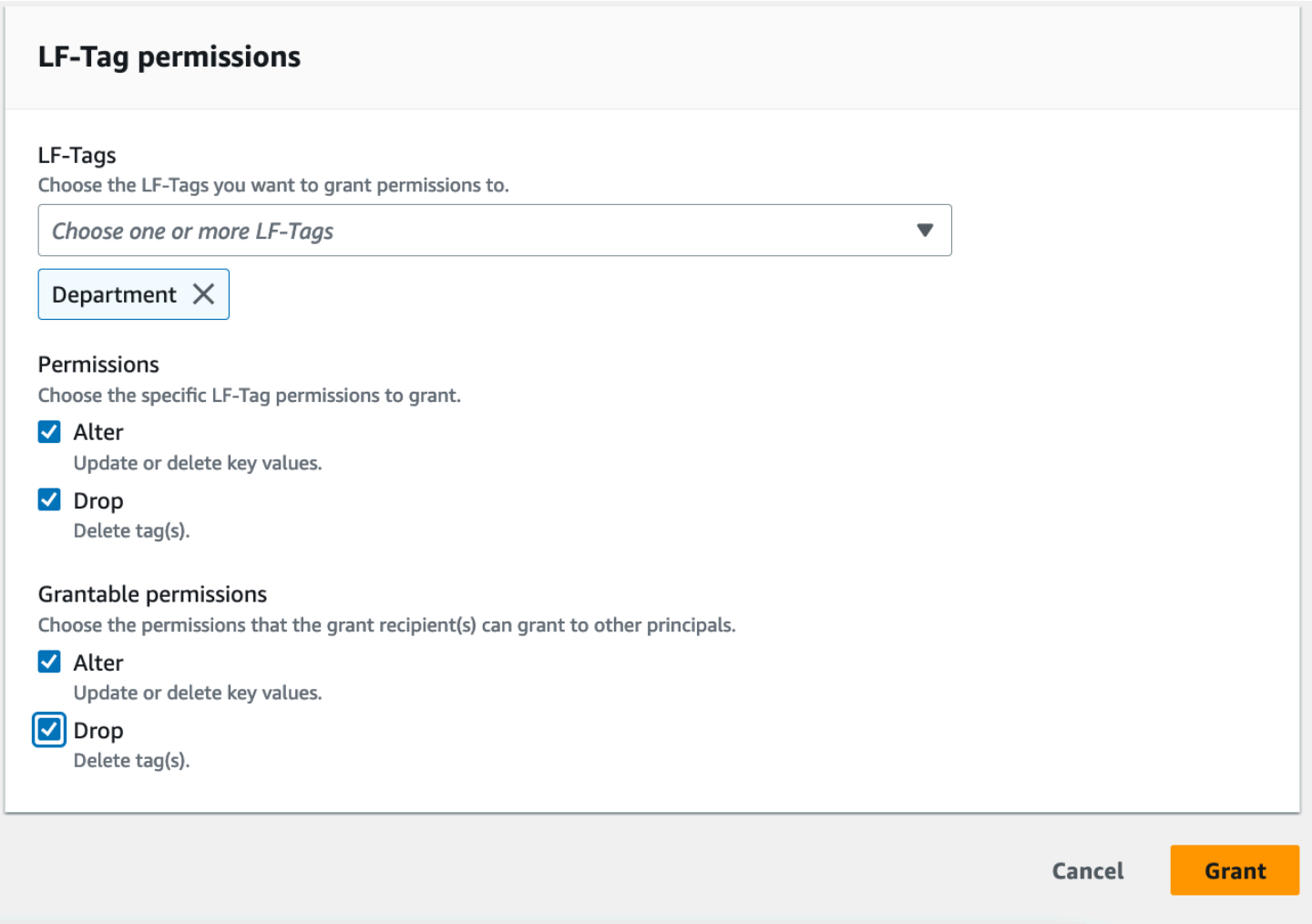
組織 ID は、最初の「o-」と、その後に続く 10 ~ 32 個の小文字または数字で構成されています。

組織単位 ID は「ou-」で始まり、その後に 4～32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」ダッシュと 8～32 個の追加の小文字または数字が続きます。

IAM プリンシパルの場合は、IAM ユーザーまたはロールの ARN を入力します。

LF タグを指定する

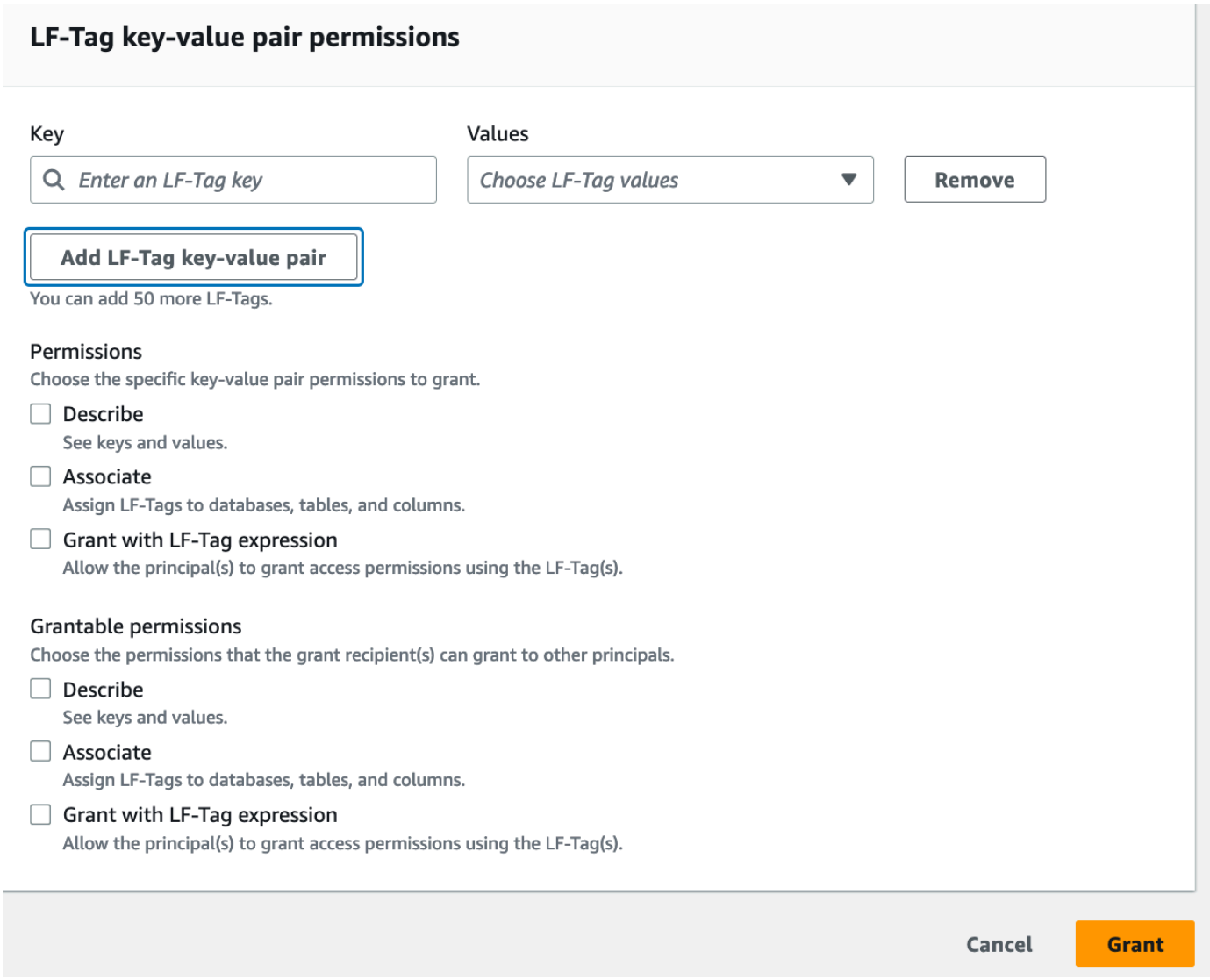
LF タグに対するアクセス許可を付与するには、[LF タグアクセス許可] セクションで、アクセス許可を付与する LF タグを指定します。



- ドロップダウンを使用して、1 つ以上の LF タグを選択します。

LF タグのキーと値のペアを指定する

1. LF タグのキーと値のペアに対するアクセス許可を付与するには (まず、[LF タグのキーと値のペアのアクセス許可] を [アクセス許可のタイプ] として選択する必要があります)、[LF タグのキーと値のペアを追加] を選択して、LF タグのキーと値を指定するフィールドの最初の行を表示します。



2. カーソルを [キー] フィールドに置き、オプションで入力を開始して選択リストを絞り込んで、LF タグのキーを選択します。

3. [Values] (値) リストで、1 つ、または複数の値を選択してから、[Tab] (タブ) を押すか、フィールドの外側をクリックまたはタップして、選択した値を保存します。

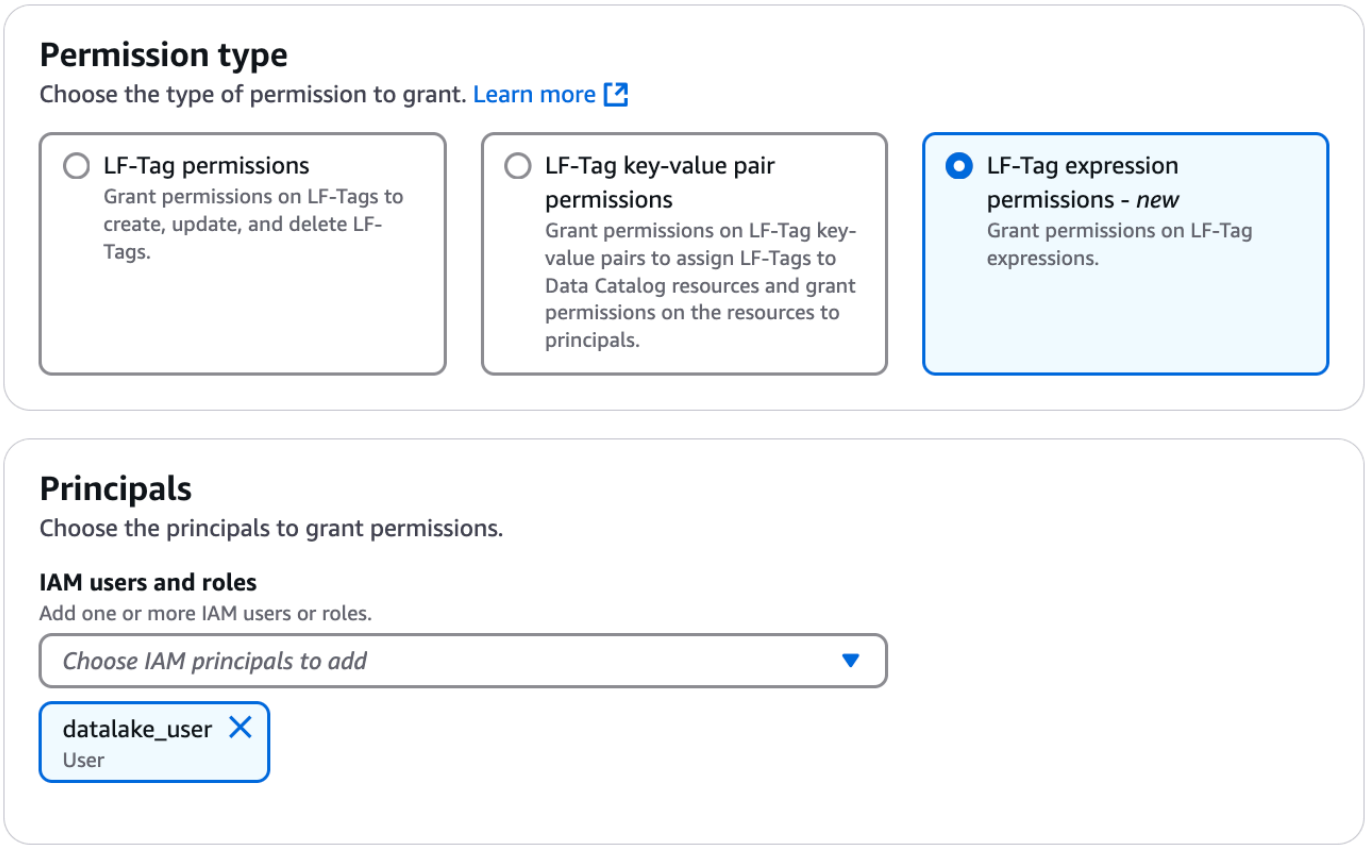
> **Note**
>
> [Values] (値) リストの行のいずれかがフォーカスされている場合は、[Enter] キーを押すと、チェックボックスがオンまたはオフになります。

選択された値は、[Values] (値) リストの下にタイルとして表示されます。✖ を選択して値を削除します。[削除] を選択して、LF タグ全体を削除します。

4. 別の LF タグを追加するには、もう一度 [LF タグを追加] を選択して、前の 2 つのステップを繰り返します。

LF タグ式を指定する

1. LF タグ式に対するアクセス許可を付与するには、(まず [LF タグ式のアクセス許可] を [アクセス許可タイプ] として選択する必要があります)。



2. LF タグ式を選択します。

3. 選択した式は、[LF タグ式] リストの下にタイルとして表示されます。✖ を選択して式を削除します。
4. 別の LF タグ式を追加するには、別の式を選択します。

許可を指定する

このセクションには、前のステップで選択した [アクセス許可タイプ] に基づいて、[LF タグアクセス許可] または [LF-タグ値のアクセス許可] のいずれかが表示されます。

付与する [アクセス許可タイプ] に応じて、[LF タグアクセス許可] または [LF タグのキーと値のペアのアクセス許可] と付与可能なアクセス許可を選択します。

1. [LF タグアクセス許可] で、付与するアクセス許可を選択します。

   [ドロップ] および [変更] を付与すると、[説明] を暗黙的に付与することになります。

   すべてのタグ値に対して、[変更] および [ドロップ] を付与する必要があります。
2. [LF タグのキーと値のペアのアクセス許可] で、付与するアクセス許可を選択します。

   [Associate] (関連付け) の付与は、[Describe] (記述) を黙示的に付与します。[LF タグ式による付与] を選択すると、付与されたユーザーは、LF-TBAC 方法を使用してデータカタログリソースに対するアクセス許可を付与または取り消すことができます。
3. [LF タグ式アクセス許可] で、付与するアクセス許可を選択します。

   [ドロップ] および [変更] を付与すると、[説明] を暗黙的に付与することになります。

   Super アクセス許可を付与すると、使用可能なすべてのアクセス許可が付与されます。
4. (オプション) 付与可能なアクセス許可で、付与受信者が AWS アカウントの他のプリンシパルに付与できるアクセス許可を選択します。
5. [付与] を選択します。

を使用した LF タグのアクセス許可の管理 AWS CLI

AWS Command Line Interface (AWS CLI) を使用して、LF タグに対するアクセス許可の付与、取り消し、および一覧表示を行うことができます。

LF タグアクセス許可を一覧表示するには (AWS CLI)

- `list-permissions` コマンドを入力します。これを表示するには、LF タグ作成者またはデータレイク管理者であるか、LF タグに対する `Drop`、`Alter`、`Describe`、`Associate`、`Grant with LF-Tag permissions` アクセス許可を持ってる必要があります。

  以下のコマンドは、アクセス許可を持っているすべての LF タグをリクエストします。

  ```
  aws lakeformation list-permissions --resource-type LF_TAG
  ```

  以下は、すべてのプリンシパルに付与されたすべての LF タグが表示される、データレイク管理者のための出力の例です。非管理ユーザーには、自分に付与された LF タグのみが表示されます。外部アカウントから付与された LF タグアクセス許可は、個別の結果ページに表示されます。それらを表示するには、コマンドの前回の実行から返されたトークンを `--next-token` 引数に指定して、コマンドを繰り返します。

  ```
  {
      "PrincipalResourcePermissions": [
          {
              "Principal": {
                  "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
  datalake_admin"
              },
              "Resource": {
                  "LFTag": {
                      "CatalogId": "111122223333",
                      "TagKey": "environment",
                      "TagValues": [
                          "*"
                      ]
                  }
              },
              "Permissions": [
                  "ASSOCIATE"
              ],
              "PermissionsWithGrantOption": [
                  "ASSOCIATE"
              ]
          },
          {
              "Principal": {
  ```

```
                "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
datalake_user1"
            },
            "Resource": {
                "LFTag": {
                    "CatalogId": "111122223333",
                    "TagKey": "module",
                    "TagValues": [
                        "Orders",
                        "Sales"
                    ]
                }
            },
            "Permissions": [
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": []
        },
...
    ],
    "NextToken": "eyJzaG91bGRRdWVy...Wlzc2lvbnMiOnRydWV9"
}
```

特定の LF タグのキーに関するすべてのアクセス許可を一覧表示できます。次のコマンドは、module という LF タグに関して付与されたすべてのアクセス許可を返します。

```
aws lakeformation list-permissions --resource-type LF_TAG --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

特定の LF タグに関して特定のプリンシパルに付与された LF タグの値を一覧表示することもできます。--principal 引数を指定する場合は、--resource 引数を指定する必要があります。このため、このコマンドが実質的にリクエストできるのは、特定の LF タグのキーに関して特定のプリンシパルに付与された値のみです。次のコマンドは、これをプリンシパル datalake_user1 と LF タグのキー module について行う方法を示しています。

```
aws lakeformation list-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1 --resource-type LF_TAG --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

以下は出力例です。

```
{
    "PrincipalResourcePermissions": [
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
datalake_user1"
            },
            "Resource": {
                "LFTag": {
                    "CatalogId": "111122223333",
                    "TagKey": "module",
                    "TagValues": [
                        "Orders",
                        "Sales"
                    ]
                }
            },
            "Permissions": [
                "ASSOCIATE"
            ],
            "PermissionsWithGrantOption": []
        }
    ]
}
```

LF タグに対するアクセス許可を付与するには (AWS CLI)

1. 以下のようなコマンドを入力します。この例は、module キーを持つ LF タグに対する Associate アクセス許可をユーザー datalake_user1 に付与します。これは、そのキーのすべての値 (アスタリスク (*) で示されています) を表示して割り当てる許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1 --permissions "ASSOCIATE" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

Associate 許可の付与は、Describe 許可を黙示的に付与します。

次の例では、キー を持つ LF タグの外部 AWS アカウント 1234-5678-9012 Associateにmoduleグラントオプションを使用して を付与します。これは、sales と orders の値のみを表示して割り当てる許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=123456789012 --permissions "ASSOCIATE"
 --permissions-with-grant-option "ASSOCIATE" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["sales", "orders"]}}'
```

2. GrantWithLFTagExpression 許可の付与は、Describe 許可を黙示的に付与します。

   次の例は、キー module を持つ LF タグに対する GrantWithLFTagExpression を付与オプション付きでユーザーに付与します。データカタログリソースを表示するアクセス許可を付与し、値 sales と orders のみを使用してアクセス許可を付与するアクセス許可を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333 --permissions "GrantWithLFTagExpression"
 --permissions-with-grant-option "GrantWithLFTagExpression" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["sales", "orders"]}}'
```

3. 次の例は、キー module を持つ LF タグに対する Drop アクセス許可を grant オプション付きでユーザーに付与します。LF タグを削除するアクセス許可を付与します。LF タグを削除するには、そのキーのすべての値に対するアクセス許可が必要です。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333 --permissions "DROP"
 --permissions-with-grant-option "DROP" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

4. 次の例は、キー module を持つ LF タグに対する Alter アクセス許可を grant オプション付きでユーザーに付与します。LF タグを削除するアクセス許可を付与します。LF タグを更新するには、そのキーのすべての値に対するアクセス許可が必要です。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=111122223333 --permissions "ALTER"
 --permissions-with-grant-option "ALTER" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

LF タグに対するアクセス許可を取り消すには (AWS CLI)

- 以下のようなコマンドを入力します。この例は、module キーを持つ LF タグに対する Associate 許可をユーザー datalake_user1 から取り消します。

```
aws lakeformation revoke-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1 --permissions "ASSOCIATE" --resource '{ "LFTag":
 {"CatalogId":"111122223333","TagKey":"module","TagValues":["*"]}}'
```

LF-TBAC 方式を使用したデータレイク許可の付与

LF タグに対する DESCRIBE と ASSOCIATE の Lake Formation 許可をプリンシパルに付与して、プリンシパルが LF タグを表示し、データカタログリソース (データベース、テーブル、ビュー、列) に割り当てるようにできます。LF タグがデータカタログリソースに割り当てられているときには、Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方法を使用して、これらのリソースをセキュリティで保護できます。詳細については、「Lake Formation のタグベースのアクセス制御」を参照してください。

最初は、データレイク管理者のみがこれらの許可を付与できます。データレイク管理者が grant オプションと共にこれらの許可を付与すると、他のプリンシパルがそれらを付与できるようになります。DESCRIBE 許可と ASSOCIATE 許可は、「Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項」で説明されています。

LF タグに対する DESCRIBEおよび アクセスASSOCIATE許可を外部 AWS アカウントに付与できます。そうすると、そのアカウントのデータレイク管理者が、アカウント内の他のプリンシパルにこれらの許可を付与できるようになります。外部アカウントのデータレイク管理者が ASSOCIATE アクセス許可を付与したプリンシパルは、アカウントを共有するデータカタログリソースに LF タグを割り当てることができます。

外部アカウントに付与するときは、grant オプションを含める必要があります。

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、LF タグに対するアクセス許可を付与できますAWS CLI。

> **Note**
>
> S3 Tables カタログでは、次の手順は必要ありません。LF タグを使用すると、既存の S3 Tables カタログを削除して再作成することなく、既存の S3 Tables カタログに対するアクセス許可を付与できます。

Lake Formation アクセス許可を使用する既存のフェデレーティッドカタログに対する LF タグのサポートを有効にする

Amazon Redshift や LF タグのサポートがフェデレーティッドカタログで利用可能になる前に作成された Amazon DynamoDB カタログなど、Lake Formation アクセス許可を使用している既存のフェデレーティッドカタログがある場合は、次の手順に従います。

1. 既存のカタログを削除する – `deleteCatalog` API オペレーションを呼び出して、Lake Formation アクセス許可を使用する既存のフェデレーティッドカタログを削除します。
2. 新しいフェデレーティッドカタログを作成する – 新しいカタログを作成し、新しいカタログを既存の名前空間/データ共有にポイントします。

   カタログに新しい名前を使用する – このプロセスでは、LF タグ機能をサポートするように既存のフェデレーティッドカタログを更新します。同じカタログ名を使用する場合は、 AWS サポートチームにお問い合わせください。

トピック

- データカタログ許可の付与

> **関連情報**
>
> - LF タグ値のアクセス許可の管理
> - メタデータアクセスコントロールのための LF タグの管理
> - Lake Formation のタグベースのアクセス制御

データカタログ許可の付与

Lake Formation コンソールまたは AWS CLI を使用して、Lake Formation タグベースのアクセスコントロール (LF-TBAC) メソッドを使用して、データカタログデータベース、テーブル、ビュー、列に対する Lake Formation アクセス許可を付与します。

Console

以下は、Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方式と Lake Formation コンソールの [データレイクのアクセス許可を付与] ページを使用して、許可を付与する方法を説明する手順です。このページは、以下のセクションに分かれています。

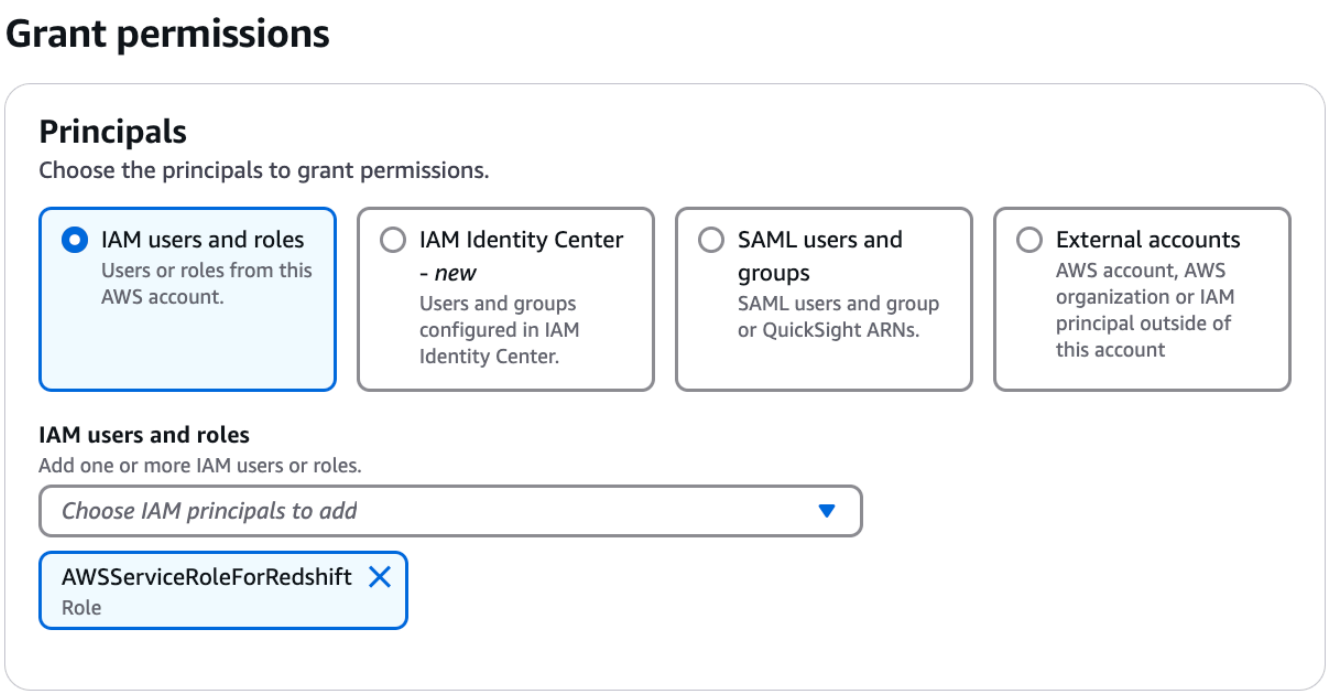
- プリンシパル – アクセス許可を付与 AWS アカウント するユーザー、ロール、および 。
- [LF-Tags or catalog resources] (LF タグまたはカタログリソース) – 付与する許可の対象となるデータベース、テーブル、またはリソースリンク。
- [Permissions] (許可) – 付与される Lake Formation 許可。

1. [データレイクのアクセス許可を付与] ページを開きます。

   https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開き、データレイク管理者として、または 許可オプションを使用して LF-TBAC を通じて Data Catalog リソースに対する Lake Formation アクセス許可を付与されたユーザーとしてサインインします。

   ナビゲーションペインの [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択します。次に、[Grant] (付与) を選択します。
2. プリンシパルを指定します。

   [Principals] (プリンシパル) セクションでプリンシパルタイプを選択してから、許可の付与先となるプリンシパルを指定します。



IAM ユーザーとロール

[IAM users and roles] (IAM ユーザーおよびロール) リストから、1 人、または複数のユーザーまたはロールを選択します。

IAM アイデンティティセンター

[ユーザーとグループ] リストから、1 人、または複数のユーザーを選択します。

SAML ユーザーとグループ

SAML およびクイックユーザーおよびグループの場合は、SAML を介してフェデレーションされたユーザーまたはグループの 1 つ以上の Amazon リソースネーム (ARNs)、またはクイックユーザーまたはグループの ARNs を入力します。各 ARN の後で Enter キーを押します。

ARN の構築方法については、「Lake Formation の許可と取り消し AWS CLI コマンド」を参照してください。

> **Note**
>
> Lake Formation と Quick の統合は、Quick Enterprise Edition でのみサポートされています。

外部アカウント

AWS アカウント、 AWS 組織、または IAM プリンシパルには、IAM ユーザーまたはロールの 1 つ以上の有効な AWS アカウント IDs、組織 IDs、組織単位 IDs、または ARN を入力します。各 ID の後で [Enter] キーを押します。

組織 ID は、最初の「o-」と、その後に続く 10 ~ 32 個の小文字または数字で構成されています。

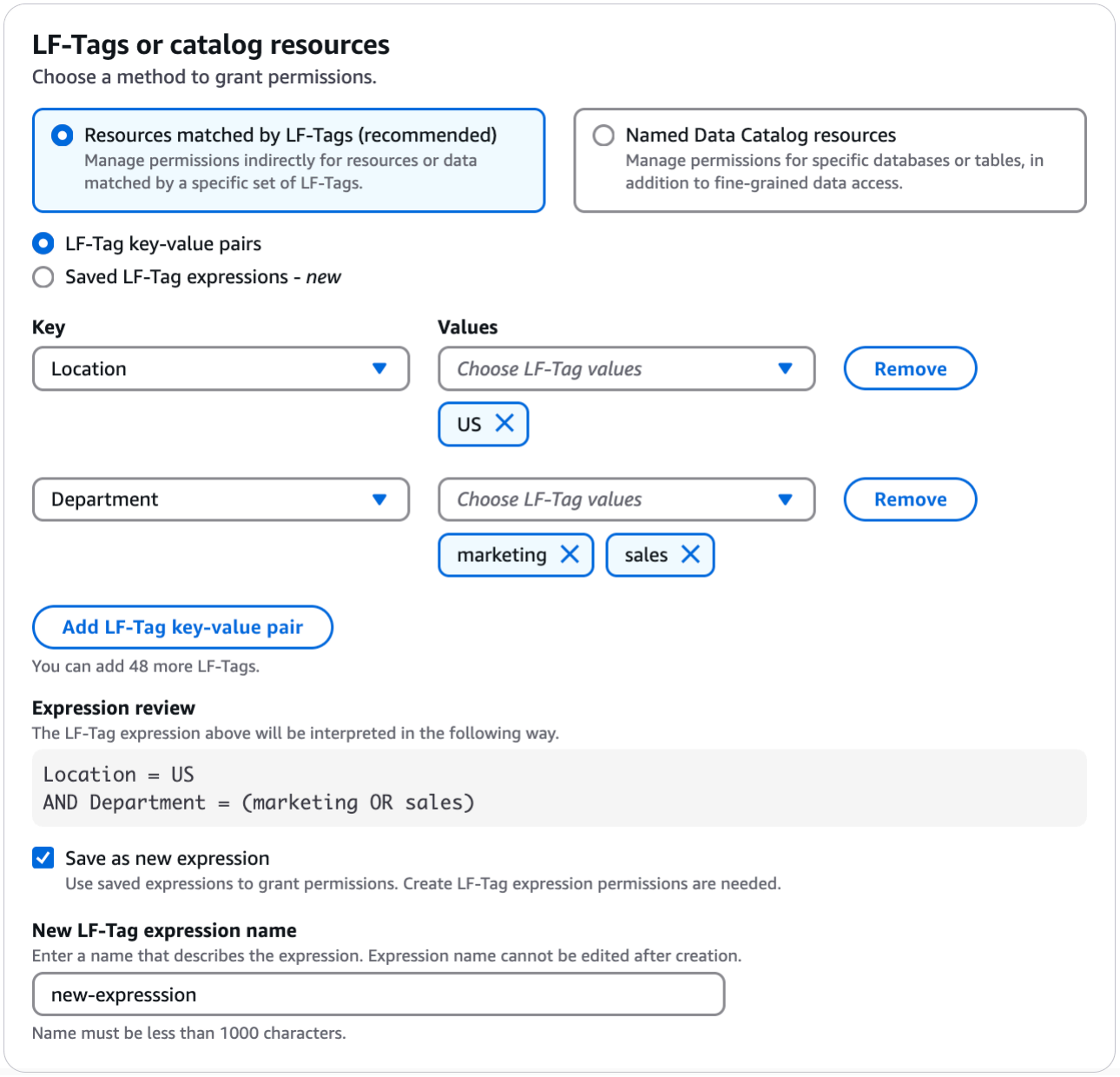
組織単位 ID は「ou-」で始まり、その後に 4 ~ 32 個の小文字または数字 (OU が含まれるルートの ID) が続きます。この文字列の後には、2 番目の「-」ダッシュと 8 ~ 32 個の追加の小文字または数字が続きます。

3. LF タグを指定します。

[LF タグに一致するリソース] オプションが選択されていることを確認します。[LF タグのキーと値のペア] または [保存された LF タグ式] を選択します。

1. [LF タグのキーと値のペア] オプションを選択した場合は、キーと値を選択します。

   複数の値を選択する場合は、OR 演算子で LF タグ式を作成することになります。これは、LF タグの値のいずれかが Data Catalog リソースに割り当てられた LF タグと一致する場合、そのリソースに対する許可が付与されることを意味します。



2. (オプション) [LF タグのキーと値のペアを追加] を再度選択して、別の LF タグを指定します。

   複数の LF タグを指定する場合は、AND 演算子で LF タグ式を作成することになります。プリンシパルには、LF タグ式内の各 LF タグに一致する LF タグが Data Catalog リソースに割り当てられている場合にのみ、その Data Catalog リソースに対する許可が付与されます。

3. 式を再利用するには、[新しい式として保存] オプションを選択します。

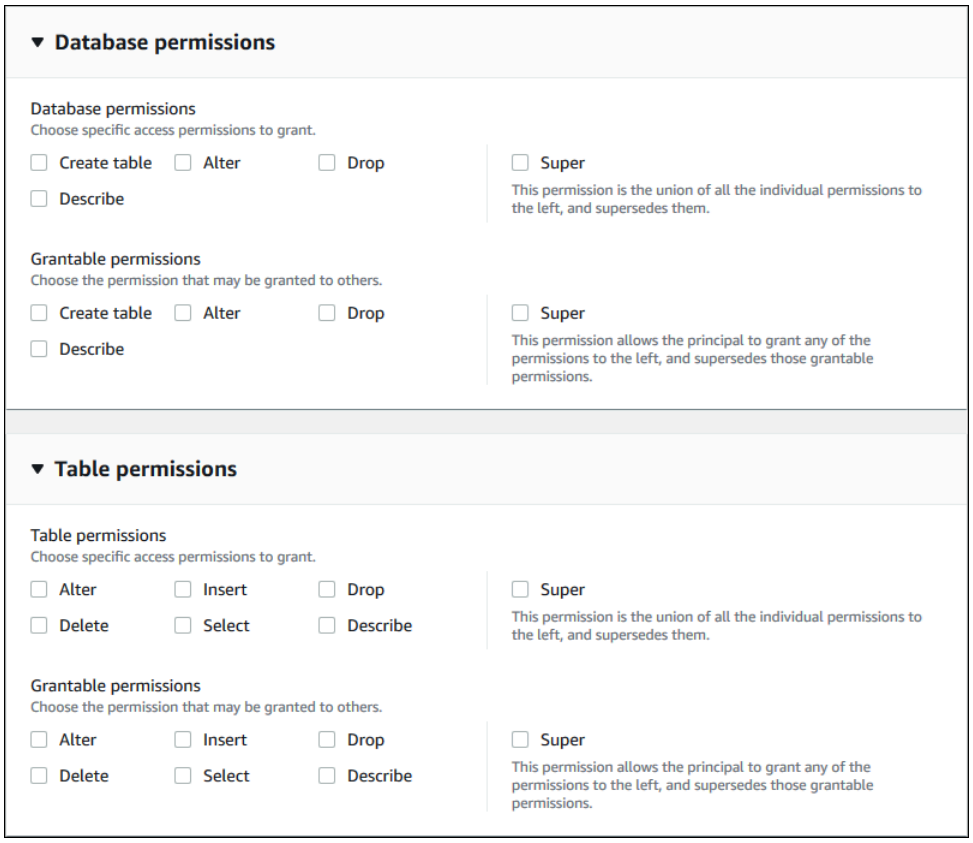
   式を保存するには `Create LF-Tag expression` が必要です。

LF タグ式に関する詳細については、「メタデータアクセスコントロールのための LF タグ式の管理」を参照してください。

4. 許可を指定します。

一致するデータカタログリソースに対してプリンシパルに付与する許可を指定します。一致するリソースとは、プリンシパルに付与された LF タグ式の 1 つに一致する LF タグが割り当てられたリソースです。

一致するデータベース、一致するテーブル、および一致するビューについて付与する許可を指定できます。



[Database permissions] (データベースの許可) で、プリンシパルに付与される、一致するデータベースに対するデータベース許可を選択します。

[テーブルの許可] で、プリンシパルに付与される、一致するテーブルとビューに対するテーブルまたはビューのアクセス許可を選択します。

[テーブルの許可] から `Select`、`Describe`、`Drop` 許可を選択してビューに適用することもできます。

5. [付与] を選択します。

AWS CLI

AWS Command Line Interface (AWS CLI) および Lake Formation タグベースのアクセスコントロール (LF-TBAC) メソッドを使用して、Data Catalog データベース、テーブル、列に対する Lake Formation アクセス許可を付与できます。

AWS CLI および LF-TBAC メソッドを使用したデータレイク許可の付与

- `grant-permissions` コマンドを実行します。

  Example

  以下の例は、LF タグ式「`module=*`」(LF タグのキー `module` のすべての値) をユーザー `datalake_user1` に付与します。このユーザーは、一致するすべてのデータベース、つまり、`module` キーと任意の値を持つ LF タグが割り当てられているデータベースに対する `CREATE_TABLE` 許可を持つようになります。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1 --permissions "CREATE_TABLE" --resource '{ "LFTagPolicy":
 {"CatalogId":"111122223333","ResourceType":"DATABASE","Expression":
[{"TagKey":"module","TagValues":["*"]}]}}'
```

  Example

  以下の例では、LF タグ式「`(level=director) AND (region=west OR region=south)`」をユーザー `datalake_user1` に付与します。このユーザーは、一致するテーブル、つまり `level=director` と (`region=west` または `region=south`) の両方が割り当てられているテーブルに対する grant オプション付きの `SELECT`、`ALTER`、および `DROP` 許可を持つようになります。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/
datalake_user1 --permissions "SELECT" "ALTER" "DROP" --permissions-
with-grant-option "SELECT" "ALTER" "DROP" --resource '{ "LFTagPolicy":
 {"CatalogId":"111122223333","ResourceType":"TABLE","Expression": [{"TagKey":
 "level","TagValues": ["director"]},{"TagKey": "region","TagValues": ["west",
 "south"]}]}}'
```

Example

次の例では、LF タグ式「」を AWS アカウント 1234-5678-9012 module=ordersに付与します。付与後、このアカウント内のデータレイク管理者は、そのアカウント内のプリンシパルに「module=orders」式を付与できるようになります。そうすると、これらのプリンシパルは、名前付きリソース方式または LF-TBAC 方式のいずれかを使用することで、アカウント 1111-2222-3333 が所有し、アカウント 1234-5678-9012 と共有されるデータベースに対する CREATE_TABLE 許可を持つようになります。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=123456789012 --permissions "CREATE_TABLE" --
permissions-with-grant-option "CREATE_TABLE" --resource '{ "LFTagPolicy":
 {"CatalogId":"111122223333","ResourceType":"DATABASE","Expression":
[{"TagKey":"module","TagValues":["orders"]}]}}'
```

属性ベースのアクセスコントロール

AWS Lake Formation では、ロールやユーザーなどの IAM エンティティに関連付けられた IAM タグとセッションタグである属性を使用して、カタログ、データベース、テーブル、データフィルターなどの AWS Glue Data Catalog オブジェクトへのアクセスを許可できます。

セッションタグの使用の詳細については、 AWS CLI ユーザーガイドの assume-role を参照してください。

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。 はこれらの属性タグを AWS 呼び出します。ABAC を使用して、同じアカウント内、またはデータカタログリソース上の別のアカウント内のプリンシパルにアクセスを許可できます。一致する IAM タグまたはセッションタグのキーと値を持つすべての IAM プリンシパルがリソースにアクセスできます。これらの許可を行うには、リソースに対する付与可能なアクセス許可が必要です。

ABAC を使用すると、複数のユーザーに同時にアクセスを許可できます。新しいユーザーが組織に加わると、管理者が特定のロールやアクセス許可を手動で割り当てなくても、ジョブ機能や部門などの属性に基づいてデータへのアクセス権を自動的に決定できます。ABAC は、ロールの代わりに属性を使用することで、さまざまなシステムや環境間でデータアクセスをより合理的で保守しやすい方法で管理できるようにし、最終的にデータガバナンスとコンプライアンスを強化します。

属性の定義の詳細については、「ABAC 認可で属性に基づいてアクセス許可を定義する」を参照してください。

制限、考慮事項、サポートされている AWS リージョンの詳細については、「」を参照してください属性ベースのアクセス制御に関する考慮事項、制限、サポートされているリージョン。

トピック

- 属性を使用してアクセス許可を付与するための前提条件
- 属性ベースのアクセス制御を使用したアクセス許可の付与

属性を使用してアクセス許可を付与するための前提条件

属性ベースのアクセス制御 (ABAC) を使用してアクセス許可を付与するには、次の前提条件を満たす必要があります。

- データカタログ設定を更新して、データカタログオブジェクトの Lake Formation アクセス許可を有効にします。詳細については、「デフォルトのアクセス許可モデルを変更する、またはハイブリッドアクセスモードを使用する」セクションを参照してください。
- クロスアカウントバージョン設定を 2 つ以上に設定します。
- アクセスを必要とする IAM エンティティに属性をアタッチします。
- データレイク管理者または必要なアクセス許可を持つ IAM ユーザーのみが、データカタログオブジェクトへのアクセスを許可できます。必要なアクセス許可の詳細については、「IAM アクセス許可」を参照してください。

属性ベースのアクセス制御を使用したアクセス許可の付与

このトピックでは、データカタログリソースに対する属性ベースのアクセス許可を付与するために必要な手順について説明します。Lake Formation コンソールまたは コマンドラインインターフェイス (AWS CLI) AWS を使用できます。

ABAC を使用したアクセス許可の付与 (AWS マネジメントコンソール)

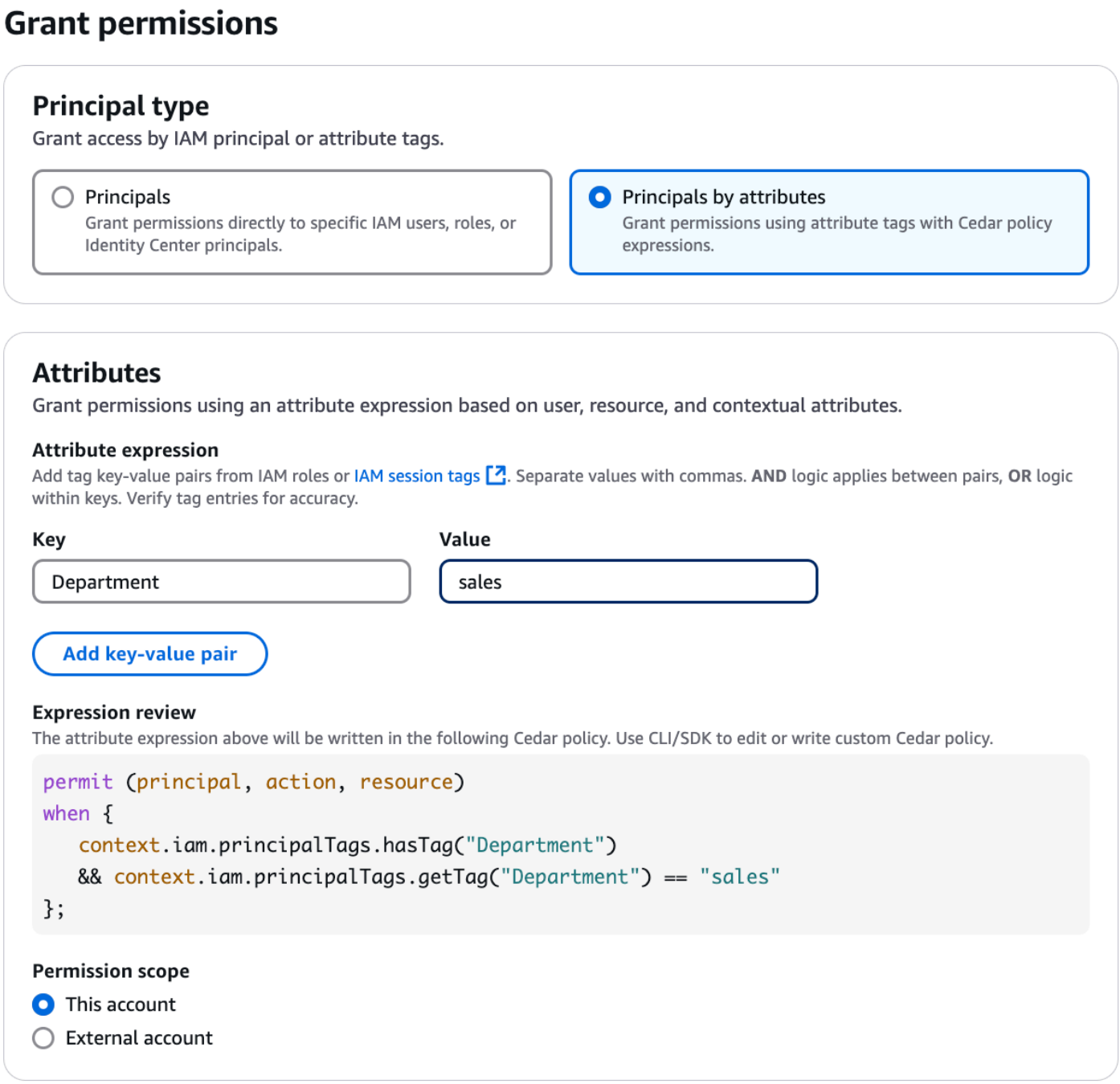
1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開き、データレイク管理者、リソース作成者、またはリソースに対して付与可能なアクセス許可を持つIAM ユーザーとしてサインインします。
2. 次のいずれかを行います。
   - ナビゲーションペインの [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択します。次に、[Grant] (付与) を選択します。

- ナビゲーションペインの[データカタログ] で、[カタログ] を選択します。次に、[アクセス許可] の [アクション] メニューからカタログオブジェクト (カタログ、データベース、テーブル、データフィルター) を選択し、[付与] を選択します。

3. 「アクセス許可の付与」ページで、[属性別プリンシパル] を選択します。
4. 属性のキーと値を指定します。複数の値を選択する場合は、OR 演算子で属性式を作成することになります。つまり、IAM ロールまたはユーザーに割り当てられた属性タグ値のいずれかが一致した場合、ロール/ユーザーはリソースに対するアクセス許可を取得します。

   複数の属性タグを指定する場合は、AND 演算子で属性式を作成することになります。プリンシパルには、属性式内の各属性タグに一致するタグが IAM ロール/ユーザーに割り当てられている場合にのみ、データカタログリソースに対する許可が付与されます。

   コンソールに表示された Cedar ポリシー式を確認します。



5. アクセス許可の範囲を選択します。被付与者が外部アカウントに属している場合は、外部アカウントを選択し、 AWS アカウント ID を入力します。
6. 次に、データカタログアカウントまたは外部アカウントを選択します。アクセス許可の付与を正常に完了するには、リソースに対応する付与可能なアクセス許可が必要です。
7. 一致する属性を持つプリンシパル (ユーザーまたはロール) に実行を許可するアクションを指定します。アクセスは、指定した属性式の少なくとも 1 つに一致するタグと値が割り当てられた IAM

エンティティに付与されます。コンソールで Cedar ポリシー式を確認します。Cedar の詳細については、「Cedar とは」を参照してください。| Cedar Policy Language Reference GuideLink を参照してください。

8. 次に、アクセスを許可するデータカタログリソースを選択します。これらのアクセス許可は、カタログ、データベース、テーブル、データフィルターなど、さまざまなデータカタログリソースに対して定義できます。
9. [付与] を選択します。

このアプローチにより、属性に基づいてアクセスを制御できるため、適切なタグを持つユーザーまたはロールのみが、指定されたリソースに対して特定のアクションを実行できます。

ABAC を使用したアクセス許可の付与 (AWS CLI)

次の例は、リソースで使用可能なすべてのアクセス許可を受け取るために満たす必要がある属性式を示しています。または、Select、Describe、Drop などの個々のアクセス許可を指定することもできます。式は Cedar ポリシー式を使用します。Cedar の詳細については、「Cedar とは」を参照してください。| Cedar Policy Language Reference GuideLink を参照してください。

この条件は、IAM プリンシパルに department タグがあり、department タグ値が sales に等しいかどうかを確認します。

```
aws lakeformation grant-permissions
--principal '{"DataLakePrincipalIdentifier": "111122223333:IAMPrincipals"}' \
--resource '{"Database": {"CatalogId": 111122223333, "Name": "abac-db"}}' \
--permissions ALL \
--condition '{"Expression": "context.iam.principalTags.hasTag(\"department\") \
   && context.iam.principalTags.getTag(\"department\") == \"sales\""}'
```

許可のシナリオ例

以下のシナリオは、 AWS Lake Formationでデータへのアクセスをセキュア化するための許可をセットアップする方法の説明に役立ちます。

Shirley はデータ管理者です。Shirley は、自分の会社である AnyCompany のためにデータレイクを設定したいと考えています。現在、すべてのデータは Amazon S3 に保存されています。John はマーケティングマネージャーで、顧客の購買情報 (s3://customerPurchases に保存されています) に対する書き込みアクセス権が必要です。この夏、マーケティングアナリストの Diego が John

の同僚になります。John には、データに対してクエリを実行するためのアクセス権を、Shirley を介さずに Diego に付与する能力が必要です。

財務部門の Mateo は、財務データ (s3://transactions など) をクエリするためのアクセス権が必要です。Mateo は、財務チームが使用しているデータベース (Finance_DB) 内のテーブルのトランザクションデータをクエリしたいと考えています。Mateo のマネージャーである Arnav は、Finance_DB へのアクセスを Mateo に許可できます。Mateo は、財務データの変更を許可されない場合でも、データを予測に適した形式 (スキーマ) に変換できる必要があります。このデータは、Mateo が変更できる別のバケット (s3://financeForecasts) に保存されます。

これを要約すると、以下のようになります。

- Shirley はデータレイク管理者です。
- John には、Data Catalog で新しいデータベースとテーブルを作成するための CREATE_DATABASE と CREATE_TABLE 許可が必要です。
- John には、作成するテーブルに対する SELECT、INSERT、および DELETE 許可も必要です。
- Diego には、クエリを実行するためのテーブルに対する SELECT 許可が必要です。

AnyCompany の従業員は、以下のアクションを実行して許可をセットアップします。このシナリオで使用される API 操作は、わかりやすくするために簡素化された構文を示しています。

1. Shirley が、顧客の購入情報が含まれる Amazon S3 パスを Lake Formation に登録します。

```
RegisterResource(ResourcePath("s3://customerPurchases"), false, Role_ARN )
```

2. Shirley が、顧客の購入情報が含まれる Amazon S3 パスへのアクセス権を John に付与します。

```
GrantPermissions(John, S3Location("s3://customerPurchases"),
 [DATA_LOCATION_ACCESS]) )
```

3. Shirley が、データベースを作成するための許可を John に付与します。

```
GrantPermissions(John, catalog, [CREATE_DATABASE])
```

4. John がデータベース John_DB を作成します。John は、データベースを作成したことから、それに対する CREATE_TABLE 許可を自動的に取得します。

```
CreateDatabase(John_DB)
```

5. John が、s3://customerPurchases をポイントするテーブル John_Table を作成します。テーブルを作成したことから、John にはテーブルに対するすべての許可があり、そのテーブルに対する許可を付与できます。

```
CreateTable(John_DB, John_Table)
```

6. John が、アナリスト Diego にテーブル John_Table へのアクセスを許可します。

```
GrantPermissions(Diego, John_Table, [SELECT])
```

7. John が、アナリスト Diego に s3://customerPurchases/London/ へのアクセスを許可します。Shirley が s3://customerPurchases を登録済みであるため、そのサブフォルダーは Lake Formation に登録されています。

```
GrantDataLakePrivileges( 123456789012/datalake, Diego, [DATA_LOCATION_ACCESS], [],
S3Location("s3://customerPurchases/London/") )
```

8. John は、アナリスト Diego に対して、データベース John_DB のテーブルを作成することを許可します。

```
GrantDataLakePrivileges( 123456789012/datalake, Diego, John_DB, [CREATE_TABLE],
[] )
```

9. Diego は John_DB のテーブルを s3://customerPurchases/London/ で作成し、ALTER、DROP、SELECTINSERT、DELETE の許可を自動的に取得します。

```
CreateTable( 123456789012/datalake, John_DB, Diego_Table )
```

Lake Formation でのデータフィルタリングとセルレベルのセキュリティ

Data Catalog テーブルに対する Lake Formation 許可を付与するときは、クエリ結果、および Lake Formation と統合されたエンジン内の特定のデータへのアクセスを制限するためのデータフィルタリング仕様を含めることができます。Lake Formation は、列レベルのセキュリティ、行レベルのセキュリティ、およびセルレベルのセキュリティを実現するために、データフィルタリングを使用します。ソースデータにネストされた構造が含まれている場合は、ネストされた列にデータフィルターを定義して適用できます。

Lake Formation のデータフィルタリング機能により、以下のレベルのデータセキュリティを実装することができます。

列レベルのセキュリティ

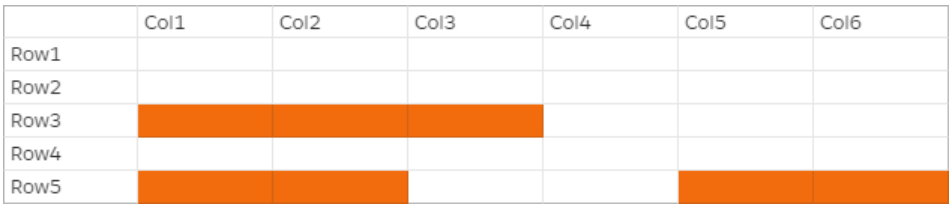
列レベルのセキュリティ (列フィルタリング) を使用して Data Catalog テーブルに対するアクセス許可を付与すると、ユーザーはそのテーブル内でアクセスが許可されている特定の列とネストされた列のみを表示できます。大規模な多地域通信会社向けの複数のアプリケーションで使用される `persons` テーブルについて考えてみましょう。Data Catalog テーブルに対する列フィルタリングを伴う許可の付与は、人事部門に属さないユーザーによる社会保障番号や生年月日などの個人を特定できる情報 (PII) の表示を制限することができます。セキュリティポリシーを定義して、ネストされた列の一部のサブ構造のみへのアクセスを許可することもできます。

行レベルのセキュリティ

Data Catalog テーブルに対する行レベルのセキュリティ (行フィルタリング) を伴う許可の付与は、ユーザーがそのテーブル内でアクセス権を持っている特定のデータの行のみを表示できるようにします。フィルタリングは、1 つ、または複数の列の値に基づいて行われます。行フィルター式を定義するときに、ネストされた列構造を含めることができます。例えば、この通信会社の異なる地域支社にそれぞれ独自の人事部門がある場合、人事部門の従業員が表示できる個人情報記録を、その地域の従業員の記録のみに制限することができます。

セルレベルのセキュリティ

セルレベルのセキュリティは、柔軟性に優れた許可モデルのために、行フィルタリングと列フィルタリングを組み合わせます。テーブルの行と列をグリッドとして考えると、セルレベルのセキュリティを使用することによって、行と列の二次元上にあれば、どこでもグリッドの個々の要素 (セル) へのアクセスを制限することができます。つまり、行に応じて異なる列へのアクセスを制限することができます。これは、制限された列に色が付けられた以下の図に表されています。



個人情報テーブルの例を引き続き使用すると、国の列が「英国」に設定されている行の住所列へのアクセスを制限するが、国の列が「米国」に設定されている行の住所列へのアクセスは許可するというデータフィルターをセルレベルで作成することができます。

フィルターは読み取り操作のみに適用されます。このため、付与できるのはフィルターを伴う SELECT Lake Formation 許可のみになります。

ネストされた列のセルレベルのセキュリティ

Lake Formationでは、ネストされた列のセルレベルのセキュリティを使用してデータフィルターを定義して適用できます。ただし、Amazon Athena、Amazon EMR、Amazon Redshift Spectrum などの統合分析エンジンは、行レベルと列レベルのセキュリティを使用した Lake Formation マネージドのネストされたテーブルに対するクエリ実行をサポートしています。

制限事項については、「データフィルタリングの制限事項」を参照してください。

トピック

- Lake Formation でのデータフィルター
- 行フィルター式での PartiQL のサポート
- セルレベルのフィルタリングを使用したテーブルのクエリに必要な許可
- データフィルターの管理

Lake Formation でのデータフィルター

データフィルターを作成することで、列レベル、行レベル、およびセルレベルのセキュリティを実装することができます。データフィルターは、テーブルに対する SELECT Lake Formation 許可を付与する時に選択します。テーブルにネストされた列構造が含まれている場合は、子列を含めるか除外するかしてデータフィルターを定義できます。また、ネストされた属性に対して行レベルのフィルター式を定義できます。

各データフィルターは、Data Catalog 内の特定のテーブルに属します。データフィルターには、以下の情報が含まれています。

- フィルター名
- フィルターが関連付けられたテーブルのカタログ ID
- テーブル名
- テーブルが含まれるデータベースの名前
- 列の指定 – クエリ結果に含めたり、クエリ結果から除外したりする列およびネストされた列 (`struct` データ型) のリスト。
- 行フィルター式 – クエリ結果に含める行を指定する式。制限はいくつかありますが、この式には PartiQL 言語の `WHERE` 句の構文があります。すべての行を指定するには、コンソールの [行レベル

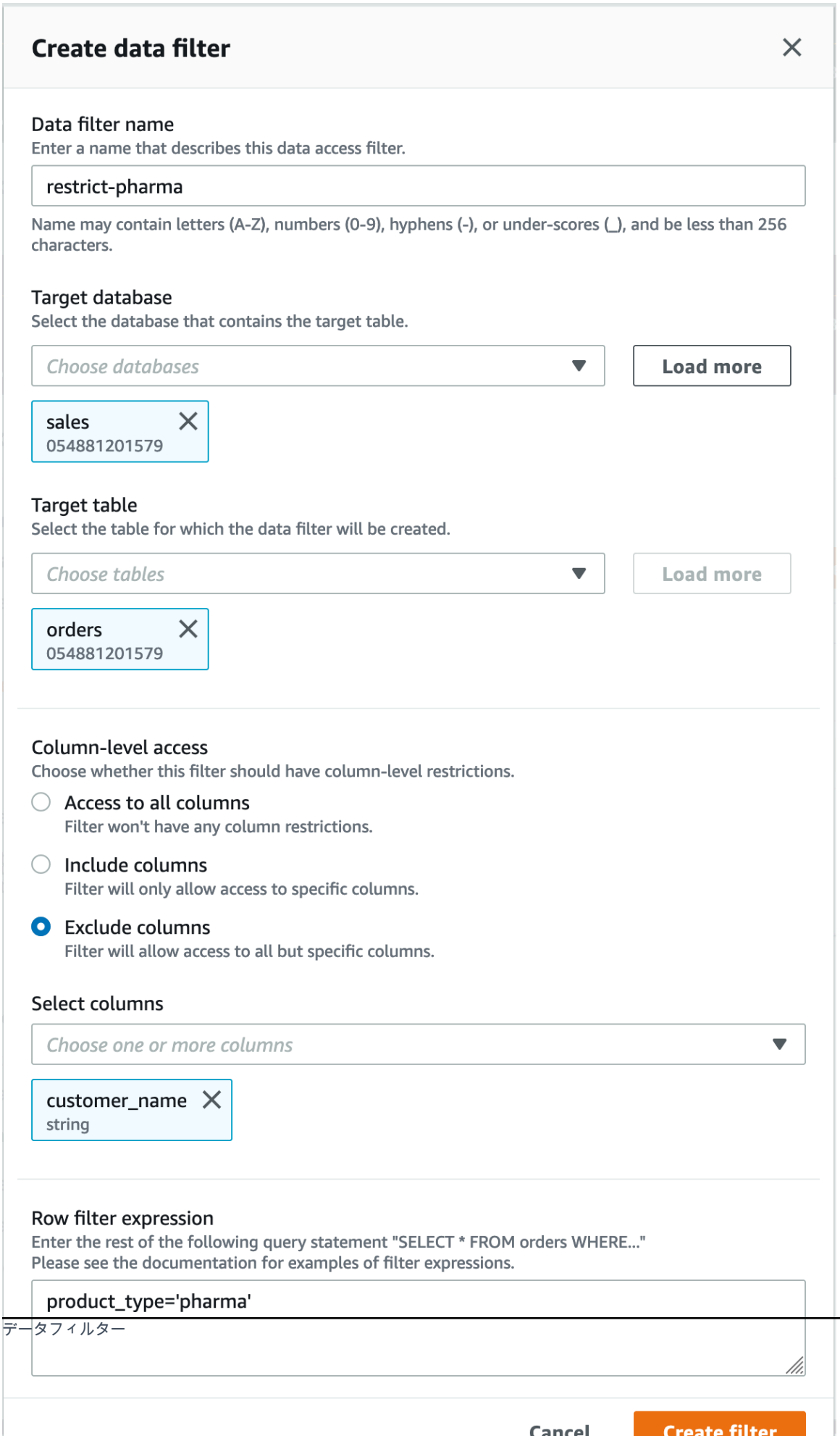
のアクセス] で [すべての行へのアクセス] を選択するか、API コールで `AllRowsWildcard` を使用します。

行フィルター式で何がサポートされるかに関する詳細については、「行フィルター式での PartiQL のサポート」を参照してください。

得られるフィルターのレベルは、データフィルターの設定方法に応じて異なります。

- 「全列」ワイルドカードを指定して、行フィルター式を提供する場合は、行レベルのセキュリティ (行フィルタリング) のみを確立することになります。
- 特定の列およびネストされた列を含めるか除外し、全行ワイルドカードを使用して「すべての行」を指定すると、列レベルのセキュリティ (列フィルタリング) のみを設定することになります。
- 特定の列を包含または除外するとともに、行フィルタリング式も指定するという場合は、セルレベルのセキュリティ (セルフィルタリング) を確立することになります。

Lake Formation コンソールからの以下のスクリーンショットは、セルレベルのフィルタリングを実行するデータフィルターを示しています。これは、`orders` テーブルに対するクエリについて、`customer_name` 列へのアクセスを制限し、クエリ結果は `product_type` 列に 'pharma' が含まれる行のみを返します。


Create data filter
Data filter name
Enter a name that describes this data access filter.
restrict-pharma
Name may contain letters (A-Z), numbers (0-9), hyphens (-), or under-scores (_), and be less than 256 characters.
Target database
Select the database that contains the target table.
Choose databases
Load more
sales
054881201579
Target table
Select the table for which the data filter will be created.
Choose tables
Load more
orders
054881201579
Column-level access
Choose whether this filter should have column-level restrictions.
Access to all columns
Filter won't have any column restrictions.
Include columns
Filter will only allow access to specific columns.
Exclude columns
Filter will allow access to all but specific columns.
Select columns
Choose one or more columns
customer_name
string
Row filter expression
Enter the rest of the following query statement "SELECT * FROM orders WHERE..."
Please see the documentation for examples of filter expressions.
product_type='pharma'
Cancel
Create filter

文字列リテラルを囲むための一重引用符の使用 ('pharma') に注意してください。

このデータフィルターを作成するには、Lake Formation コンソールを使用するか、CreateDataCellsFilter API 操作に以下のリクエストオブジェクトを提供することができます。

```
{
     "Name": "restrict-pharma",
     "DatabaseName": "sales",
     "TableName": "orders",
     "TableCatalogId": "111122223333",
     "RowFilter": {"FilterExpression": "product_type='pharma'"},
     "ColumnWildcard": {
         "ExcludedColumnNames": ["customer_name"]
     }
}
```

テーブルには、必要な数だけデータフィルターを作成できます。これには、テーブルに対する grant オプション付きの SELECT 許可が必要です。データレイク管理者にはデフォルトで、そのアカウント内のすべてのテーブルに対してデータフィルターを作成するアクセス許可があります。通常、テーブルに対する許可をプリンシパルに付与するときは、使用可能なデータフィルターのサブセットのみを使用します。例えば、orders テーブルのために、行セキュリティのみのデータフィルターである 2 番目のデータフィルターを作成することができます。上記のスクリーンショットを参考にすると、[Access to all columns] (すべての列にアクセス) オプションを選択して、product_type<>pharma という行フィルター式を含めることができます。このデータフィルターの名前は no-pharma にすることができます。これは、product_type 列が 'pharma' に設定されているすべての行に対するアクセスを制限します。

以下は、このデータフィルターの CreateDataCellsFilter API 操作のリクエストオブジェクトです。

```
{
     "Name": "no-pharma",
     "DatabaseName": "sales",
     "TableName": "orders",
     "TableCatalogId": "111122223333",
     "RowFilter": {"FilterExpression": "product_type<>'pharma'"},
     "ColumnNames": ["customer_id", "customer_name", "order_num"
          "product_id", "purchase_date", "product_type",
          "product_manufacturer", "quantity", "price"]
```

```
}
```

その後、orders テーブルに対する restrict-pharma データフィルターを伴う SELECT を管理者ユーザーに、orders テーブルに対する no-pharma データフィルターを伴う SELECT を非管理者ユーザーに付与することができます。ヘルスケア部門のユーザーの場合は、orders テーブルに対するすべての行と列への完全なアクセス権を伴う (データフィルターなし) SELECT を付与するか、料金設定情報へのアクセスを制限する別のデータフィルターを使用するものを付与こともできます。

データフィルター内に列レベルと行レベルのセキュリティを指定する際に、ネストされた列を含めるか除外することができます。次の例では、修飾列名 (二重引用符で囲まれた列名) を使用して product.offer フィールドへのアクセスを指定しています。これは、ネストされたフィールドにとって、列名に特殊文字が含まれている場合にエラーが発生するのを防ぎ、最上位の列レベルのセキュリティ定義との下位互換性を維持するために重要です。

```
{
     "Name": "example_dcf",
     "DatabaseName": "example_db",
     "TableName": "example_table",
     "TableCatalogId": "111122223333",
     "RowFilter": { "FilterExpression": "customer.customerName <> 'John'" },
     "ColumnNames": ["customer", "\"product\".\"offer\""]
}
```

以下も参照してください。

- データフィルターの管理

行フィルター式での PartiQL のサポート

PartiQL データ型、演算子、および集計のサブセットを使用して、行フィルター式を構築することができます。Lake Formation では、フィルター式にユーザー定義または標準の PartiQL 関数は使用できません。比較演算子を使用して、列を定数 (例えば `views >= 10000`) と比較することはできますが、列を他の列と比較することはできません。

行フィルター式は、単純式または複合式にすることができます。式の合計長は 2048 文字未満にする必要があります。

単純式

単純式は、次の形式になります: `<column name > <comparison operator ><value >`

- Column name (列名)

  これは、テーブルスキーマに存在する最上位レベルのデータ列、パーティション列、またはネストされた列のいずれかであり、以下に示すサポートされているデータ型に属している必要があります。

- Comparison operator (比較演算子)

  サポートされている演算子は、次のとおりです: `=, >, <, >=, <=, <>,!=, BETWEEN, IN, LIKE, NOT, IS [NOT] NULL`

- すべての文字列比較および `LIKE` パターンマッチングでは、大文字と小文字が区別されます。IS [NOT] NULL 演算子は、パーティション列には使用できません。
- Column value (列値)

  列値は、列名のデータ型に一致する必要があります。

複合式

複合式は、次の形式になります: `( <simple expression >) <AND/OR >(<simple expression >)` 複合式は、論理演算子 `AND/OR` を使用してさらに組み合わせることができます。

サポートされているデータ型

サポートされていないデータ型を含む AWS Glue Data Catalog テーブルを参照する行フィルターはエラーになります。以下は、データ型にマッピングされるテーブル列と定数でサポートされている Amazon Redshift データ型です。

- `STRING, CHAR, VARCHAR`
- `INT, LONG, BIGINT, FLOAT, DECIMAL, DOUBLE`
- `BOOLEAN`
- `STRUCT`

Amazon Redshift のデータ型の詳細については、「Amazon Redshift データベースデベロッパーガイド」の「データ型」を参照してください。

行フィルター式

Example

以下は、次の列を持つテーブルに対する有効な行フィルター式の例です: `country (String), id (Long), year (partition column of type Integer), month (partition column of type Integer)`

- `year > 2010 and country != 'US'`
- `(year > 2010 and country = 'US') or (month < 8 and id > 23)`
- `(country between 'Z' and 'U') and (year = 2018)`
- `(country like '%ited%') and (year > 2000)`

Example

ネストされた列を持つテーブルに対する有効な行フィルター式の例は、次のとおりです: `year > 2010 and customer.customerId <> 1`

ネストされた行レベルの式を定義するときは、パーティション列の下のネストされたフィールドを参照しないでください。

文字列定数は一重引用符で囲む必要があります。

予約キーワード

行フィルター式に PartiQL キーワードが含まれている場合、列名がキーワードと競合する可能性があることから構文解析エラーが発生します。このエラーが発生した場合は、二重引用符を使用して列名をエスケープしてください。予約キーワードの例には、「first」、「last」、「asc」、「missing」などがあります。予約キーワードのリストについては、PartiQL の仕様を参照してください。

PartiQL リファレンス

PartiQL の詳細については、https://partiql.org/ を参照してください。

セルレベルのフィルタリングを使用したテーブルのクエリに必要な許可

セルレベルのフィルタリングを使用してテーブルに対してクエリを実行するには、次の AWS Identity and Access Management (IAM) アクセス許可が必要です。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:StartQueryPlanning",
                "lakeformation:GetQueryState",
                "lakeformation:GetWorkUnits",
                "lakeformation:GetWorkUnitResults"
            ],
            "Resource": "*"
        }
    ]
}
```

Lake Formation の許可の詳細については、「Lake Formation のペルソナと IAM 許可のリファレンス」を参照してください。

データフィルターの管理

列レベル、行レベル、およびセルレベルのセキュリティを実装するには、データフィルターを作成して維持することができます。各データフィルターは、Data Catalog テーブルに属します。テーブル用に複数のデータフィルターを作成してから、そのテーブルに対する許可を付与するときに 1 つ、または複数のデータフィルターを使用できます。また、`struct` データ型を持つネストされた列にデータフィルターを定義して適用し、ネストされた列のサブ構造のみへのアクセスをユーザーに許可することもできます。

データフィルターを作成または表示するには、grant オプション付きの `SELECT` 許可が必要です。アカウントのプリンシパルがデータフィルターを表示して使用できるようにするには、そのデータフィルターに対する `DESCRIBE` 許可を付与することができます。

> **Note**
>
> Lake Formation は、別のアカウントから共有されているデータフィルターへの `Describe` アクセス許可の付与をサポートしていません。

データフィルターは、 AWS Lake Formation コンソール、 API、または AWS Command Line Interface () を使用して管理できますAWS CLI。

データフィルターについては、「Lake Formation でのデータフィルター」を参照してください。

データフィルターの作成

Data Catalog テーブルごとに、1 つ、または複数のデータフィルターを作成できます。

Data Catalog テーブルのデータフィルターを作成する (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、ターゲットテーブル所有者、またはターゲットテーブルに対する Lake Formation 許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインの [Data catalog] で [Data filters] (データフィルター) を選択します。
3. [Data filters] (データフィルター) ページで、[Create new filter] (新しいフィルターを作成) を選択します。
4. [Create data filter] (データフィルターの作成) ダイアログボックスで、以下の情報を入力します。

   - [Data filter name] (データフィルター名)
   - [Target database] (ターゲットデータベース) – テーブルが含まれるデータベースを指定します。
   - [Target table] (ターゲットテーブル)
   - [Column-level access] (列レベルのアクセス) – 行フィルターのみを指定する場合は、[Access to all columns] (すべての列にアクセス) のままにしておきます。列またはセルフィルタリングを指定する場合は、[Include columns] (列を含める) または[Exclude columns] (列を除外する) を選択してから、含める列、または除外する列を指定します。

     ネストされた列 — ネストされた列を含むテーブルにフィルターを適用する場合、データフィルター内でネストされた構造体列のサブ構造を明示的に指定できます。

     このフィルターでプリンシパルに SELECT アクセス許可を付与すると、次のクエリを実行するプリンシパルには、`customer.customerName` のデータのみが表示され、`customer.customerId` のデータは表示されません。

     ```
     SELECT "customer" FROM "example_db"."example_table";
     ```

Column-level access

Choose whether this filter should have column-level restrictions.

Column-level access

Choose whether this filter should have column-level restrictions.

- ○ Access to all columns
  Filter won't have any column restrictions.
- ◉ Include columns
  Filter will only allow access to specific columns.
- ○ Exclude columns
  Filter will allow access to all but specific columns.

Included columns (4/11)

Choose the columns for column-level access

Find column

1

Name	Type
☐ customer	struct
☐ customerId	string
☑ customerName	string
☑ customerapplication	struct
☐ appId	string
☑ product	struct
☐ offer	struct
☐ listingId	string
☐ prodId	string
☐ type	string
☑ purchaseid	string

Row-level access

Choose whether this filter should have row-level restrictions.

- ○ Access to all rows
- ◉ Filter rows

Row filter expression

Enter the rest of the following query statement SELECT * FROM nested-table WHERE...
Please see the documentation for examples of filter expressions.

```
customer.customerName <> 'John'
```

customer 列にアクセス許可を付与すると、プリンシパルは、列とその列の下にネストされたフィールド (customerName と customerID) へのアクセス権を受け取ります。

- [Row filter expression] (行フィルター式) – 行またはセルフィルタリングを指定するフィルター式を入力します。サポートされるデータ型と演算子については、「行フィルター式での PartiQL のサポート」を参照してください。[すべての行へのアクセス] を選択して、すべての行に対するアクセスを許可します。

  ネストされた列の一部の列構造を行フィルター式に含めて、特定の値を含む行をフィルターできます。

  行フィルター式 `Select * from example_nestedtable where customer.customerName <>'John'` を使用してテーブルに対するアクセス許可をプリンシパルに付与し、列レベルのアクセスを [すべての列へのアクセス] に設定すると、クエリ結果には `customerName <>'John'` が true と評価された行のみが表示されます。

次のスクリーンショットは、セルフィルタリングを実装するデータフィルターを示しています。orders テーブルに対するクエリでは、customer_name 列へのアクセスが拒否され、product_type 列に 'pharma' がある行のみが表示されます。

Create data filter

Data filter name
Enter a name that describes this data access filter.

restrict-pharma

Name may contain letters (A-Z), numbers (0-9), hyphens (-), or under-scores (_), and be less than 256 characters.

Target database
Select the database that contains the target table.

Choose databases | Load more

sales
054881201579

Target table
Select the table for which the data filter will be created.

Choose tables | Load more

orders
054881201579

Column-level access
Choose whether this filter should have column-level restrictions.

- Access to all columns
  Filter won't have any column restrictions.
- Include columns
  Filter will only allow access to specific columns.
- Exclude columns
  Filter will allow access to all but specific columns.

Select columns

Choose one or more columns

customer_name
string

5. [Create filter] (フィルターを作成) を選択します。

ネストされたフィールドでセルフィルターポリシーを使用してデータフィルターを作成するには

このセクションでは、次のサンプルスキーマを使用してデータセルフィルターを作成する方法を示します。

```
[
    { name: "customer", type: "struct<customerId:string,customerName:string>" },
    { name: "customerApplication", type: "struct<appId:string>" },
    { name: "product", type:
 "struct<offer:struct<prodId:string,listingId:string>,type:string>" },
    { name: "purchaseId", type: "string" },
]
```

1. [データフィルターを作成] ページで、データフィルターの名前を入力します。
2. 次に、ドロップダウンを使用してデータベース名とテーブル名を選択します。
3. [列レベルのアクセス] セクションで、[含まれる列] を選択し、ネストされた列 (`customer.customerName`) を選択します。
4. [行レベルのアクセス] セクションで、[すべての行へのアクセス] オプションを選択します。
5. [フィルターを作成] をクリックします。

   このフィルターで SELECT アクセス許可を付与すると、プリンシパルは `customerName` 列内のすべての行にアクセスできるようになります。
6. 次に、同じデータベース/テーブルに別のデータフィルターを定義します。
7. [列レベルのアクセス] セクションで、[含まれる列] を選択し、別のネストされた列 (`customer.customerid`) を選択します。
8. [行レベルのアクセス] セクションで、[行をフィルタリングする] を選択し、[行フィルター式] (`customer.customerid <> 5`) を入力します。
9. [フィルターを作成] をクリックします。

   このフィルターで SELECT アクセス許可を付与すると、プリンシパルは、`customerName` フィールドと `customerId` フィールド (`customerId` 列の値が 5 であるセルを除く) のすべての行にアクセスできるようになります。

データフィルターの許可の付与

プリンシパルには、データフィルターに対する SELECT、DESCRIBE、および DROP Lake Formation 許可を付与することができます。

当初、テーブル用に作成したデータフィルターを表示できるのは、作成したユーザーだけです。別のプリンシパルがデータフィルターを表示して、そのデータフィルターを伴う Data Catalog 許可を付与できるようにするには、以下のいずれかを実行する必要があります。

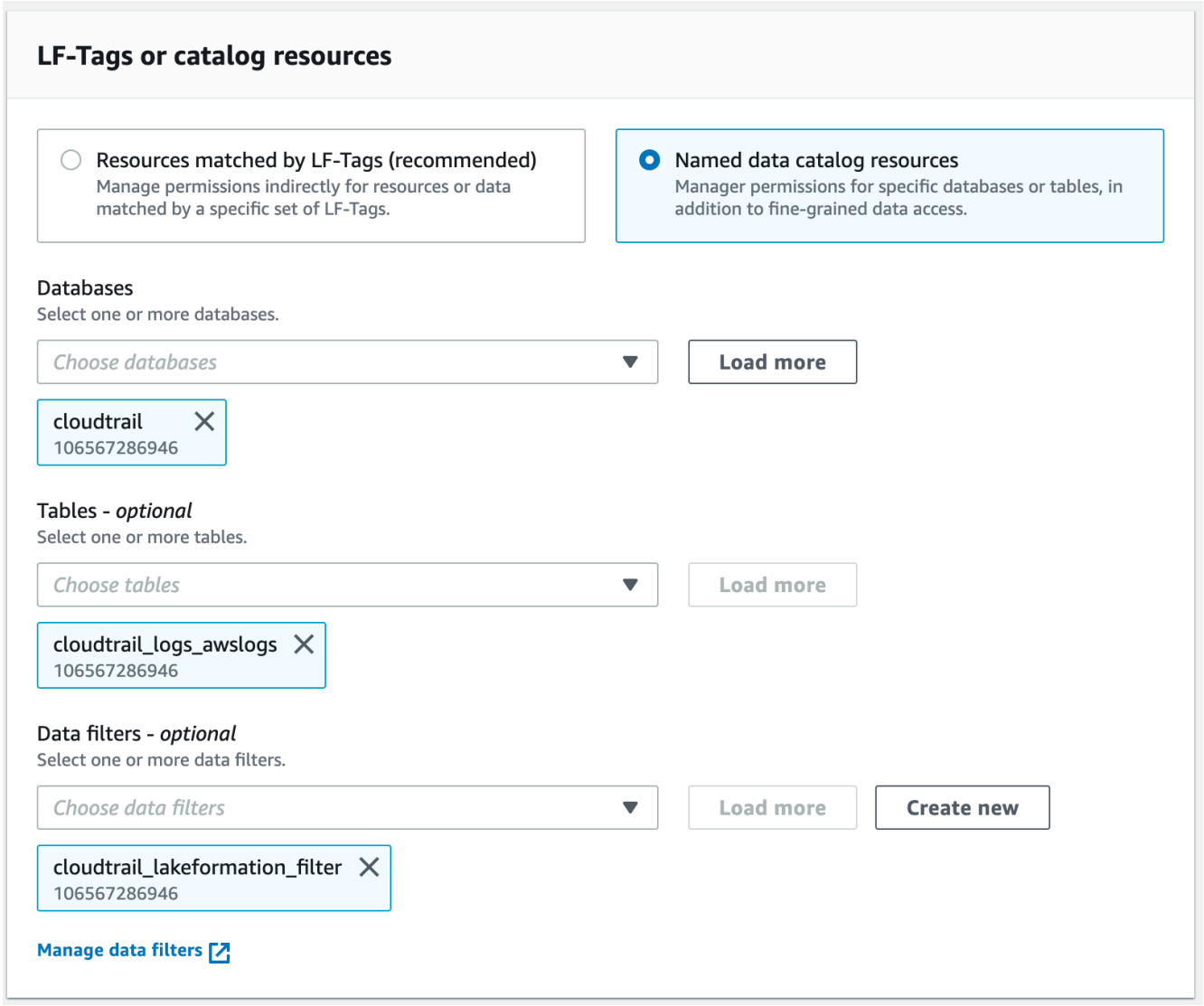
- テーブルに対する grant オプション付きの SELECT をプリンシパルに付与し、その付与にデータフィルターを適用する。
- データフィルターに対する DESCRIBE または DROP 許可をプリンシパルに付与する。

外部 AWS アカウントに アクセスSELECT許可を付与できます。付与後、そのアカウントのデータレイク管理者は、アカウント内の他のプリンシパルにその許可を付与できるようになります。外部アカウントに付与するときは、外部アカウントの管理者がそのアカウント内の他のユーザーに許可をさらにカスケードできるように、grant オプションを含める必要があります。アカウント内のプリンシパルに付与するときの grant オプションを伴う付与はオプションです。

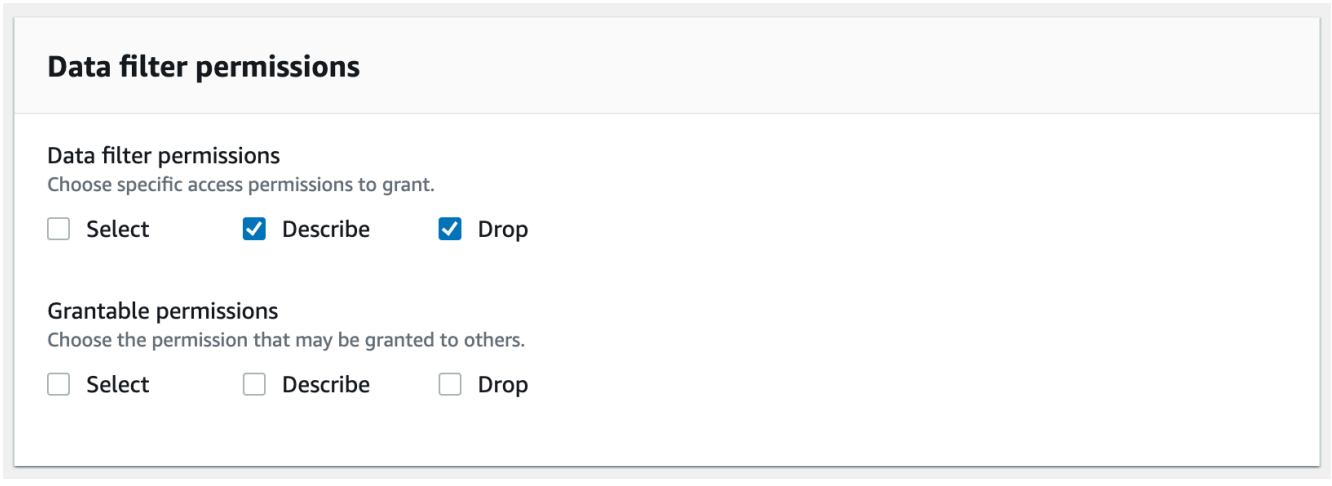
AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、データフィルターに対するアクセス許可を付与および取り消すことができますAWS CLI。

Console

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/lakeformation/ で Lake Formation コンソールを開きます。
2. ナビゲーションペインの [Permissions] (許可) で [Data lake permissions] (データレイクの許可) を選択します。
3. [Permissions] (許可) ページの [Data permissions] (データの許可) セクションで、[Grant] (付与) を選択します。
4. [Grant data permissions] (データ許可の付与) ページで、許可を付与するプリンシパルを選択します。
5. [LF-Tags or catalog resources] (LF タグまたはカタログリソース) セクションで、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。次に、許可を付与するデータベース、テーブル、およびデータフィルターを選択します。



6. [Data filter permissions] (データフィルターの許可) セクションで、選択したプリンシパルに付与する許可を選択します。



AWS CLI

- grant-permissions のコマンドを入力します。resource 引数に DataCellsFilter を指定し、Permissions 引数、およびオプションで PermissionsWithGrantOption 引数に、DESCRIBE または DROP を指定します。

  以下の例は、データフィルター restrict-pharma (AWS アカウント 1111-2222-3333 内の sales データベースにある orders テーブルの属するもの) に対する grant オプション付きの DESCRIBE をユーザー datalake_user1 に付与します。

```
aws lakeformation grant-permissions --cli-input-json file://grant-params.json
```

  以下は、ファイル grant-params.json の内容です。

```
{
    "Principal": {"DataLakePrincipalIdentifier":
 "arn:aws:iam::111122223333:user/datalake_user1"},
    "Resource": {
        "DataCellsFilter": {
            "TableCatalogId": "111122223333",
            "DatabaseName": "sales",
            "TableName": "orders",
            "Name": "restrict-pharma"
        }
    },
    "Permissions": ["DESCRIBE"],
    "PermissionsWithGrantOption": ["DESCRIBE"]
}
```

データフィルターが提供するデータの許可の付与

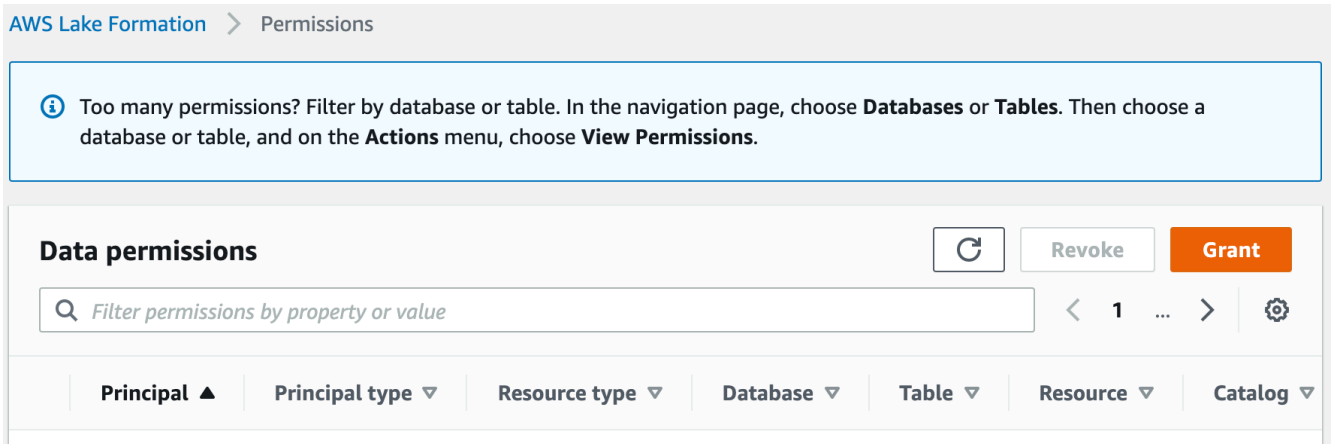
データフィルターは、テーブル内のデータのサブセットを表します。プリンシパルにデータアクセスを提供するには、これらのプリンシパルに SELECT 許可を付与する必要があります。この許可により、プリンシパルは以下を実行できます。

- プリンシパルのアカウントと共有されているテーブルのリストで実際のテーブル名を表示する。
- 共有テーブルでデータフィルターを作成し、これらのデータフィルターに対する許可をユーザーに付与します。

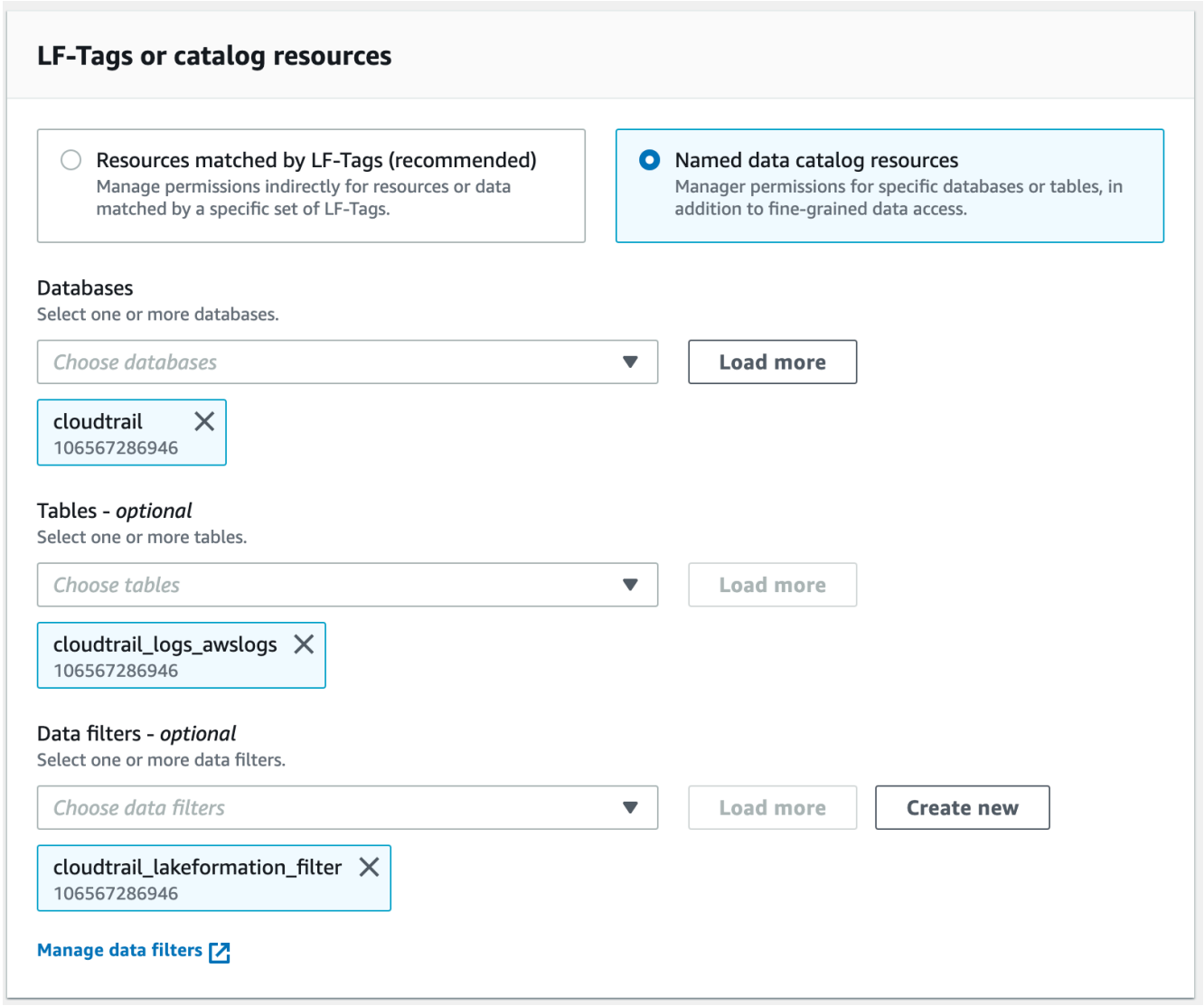
Console

SELECT 許可を付与する

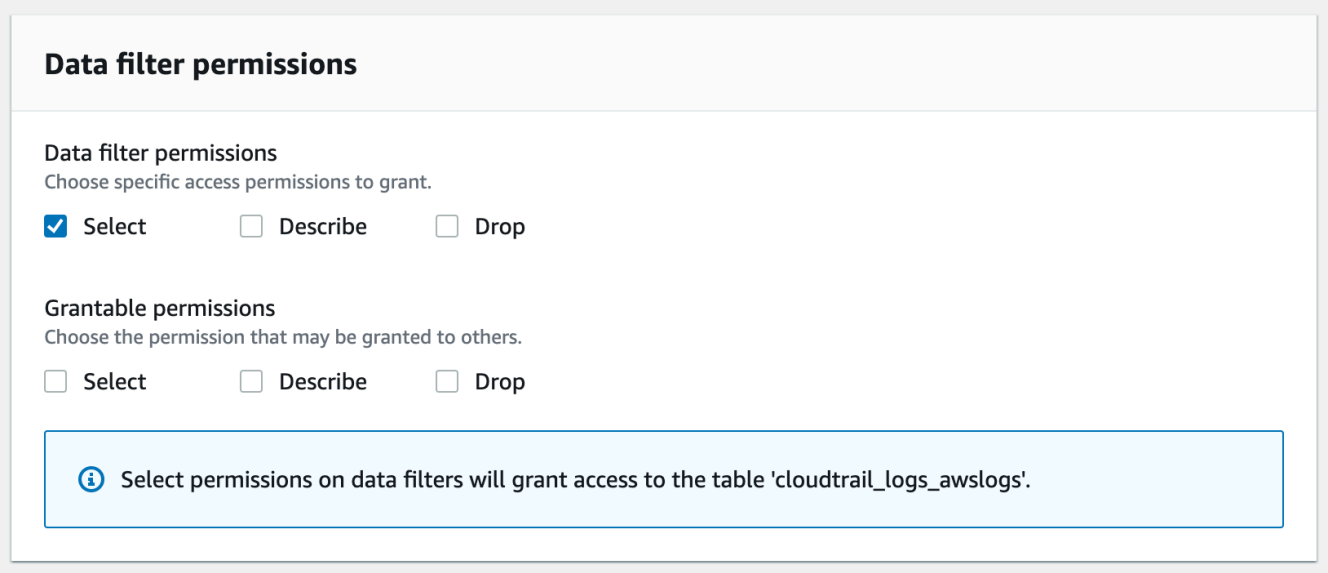
1. Lake Formation コンソールで [Permissions] (許可) ページに移動し、[Grant] (付与) を選択します。



2. アクセス権を付与する先のプリンシパルを選択し、[Named data catalog resources] (名前付きの Data Catalog リソース) を選択します。



3. フィルターが表すデータへのアクセス権を提供するには、[Data filter permissions] (データフィルターの許可) で [Select] (選択) を選択します。



CLI

grant-permissions のコマンドを入力します。Resource 引数に DataCellsFilter を指定し、Permissions 引数に SELECT を指定します。

次の例では、 のsalesデータベースの ordersテーブルに属するデータフィルターdatalake_user1でrestrict-pharma、 付与オプションSELECTを使用して をユーザーに付与します AWS アカウント 1111-2222-3333。

```
aws lakeformation grant-permissions --cli-input-json file://grant-params.json
```

以下は、ファイル grant-params.json の内容です。

```
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
datalake_user1"
    },
    "Resource": {
        "DataCellsFilter": {
            "TableCatalogId": "111122223333",
            "DatabaseName": "sales",
            "TableName": "orders",
            "Name": "restrict-pharma"
        }
    },
```

```
    "Permissions": ["SELECT"]
}
```

データフィルターの表示

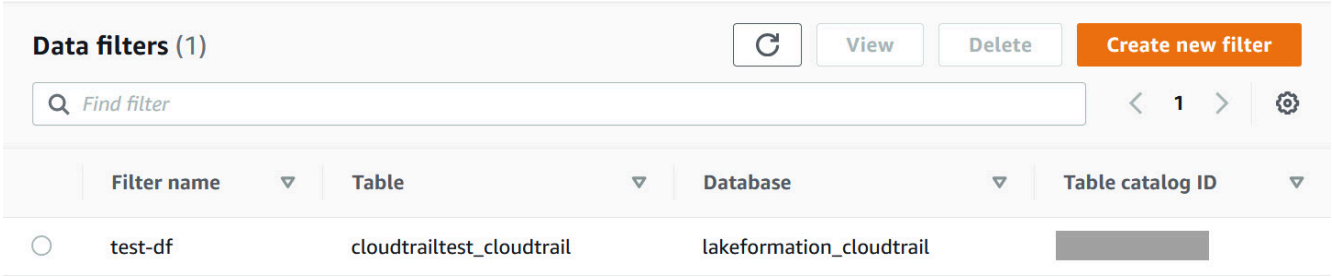
Lake Formation コンソール AWS CLI、または Lake Formation API を使用して、データフィルターを表示できます。

データフィルターを表示するには、Data Lake 管理者であるか、データフィルターに対する必要な許可を持っている必要があります。

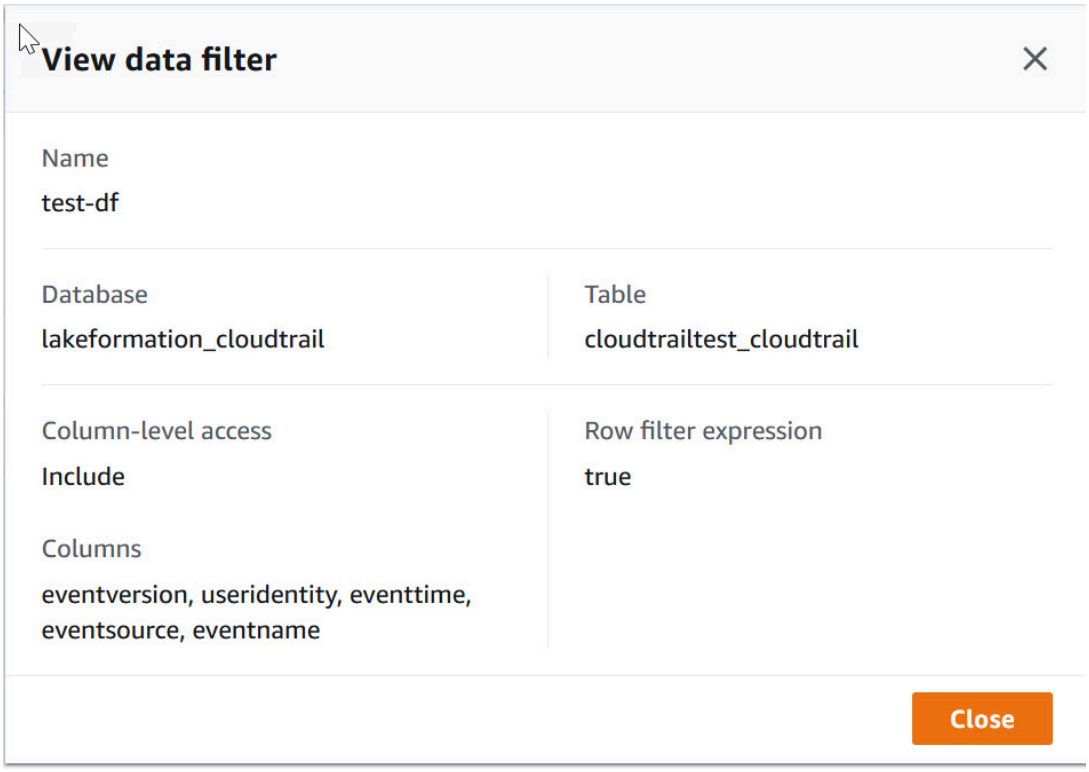
Console

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/lakeformation/ で Lake Formation コンソールを開きます。
2. ナビゲーションペインの [Data catalog] で [Data filters] (データフィルター) を選択します。

   このページには、アクセスできるデータフィルターが表示されます。



3. データフィルターの詳細を表示するには、データフィルターを選択してから [View] (表示) を選択します。データフィルターの詳細情報が記載された新しいウィンドウが開きます。



AWS CLI

`list-data-cells-filter` コマンドを入力して、テーブルリソースを指定します。

以下の例は、`cloudtrailtest_cloudtrail` テーブルのデータフィルターをリストします。

```
aws lakeformation list-data-cells-filter --table '{ "CatalogId":"123456789012",
"DatabaseName":"lakeformation_cloudtrail", "Name":"cloudtrailtest_cloudtrail"}'
```

API/SDK

`ListDataCellsFilter` API を使用して、テーブルリソースを指定します。

以下の例は、Python を使用して `myTable` テーブルの最初 20 個のデータフィルターをリストします。

```
response = client.list_data_cells_filter(
    Table = {
        'CatalogId': '111122223333',
        'DatabaseName': 'mydb',
        'Name': 'myTable'
```

```
    },
    MaxResults=20
)
```

データフィルターの許可の表示

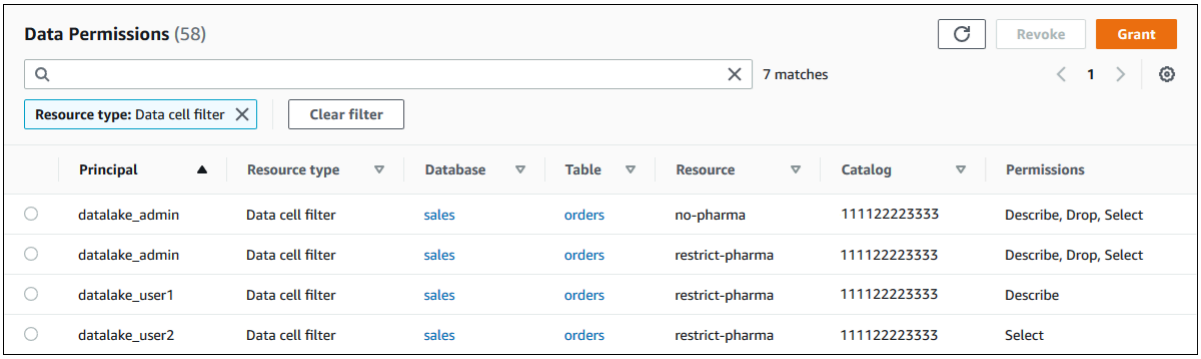
Lake Formation コンソールを使用して、データフィルターに対して付与された許可を表示できます。

データフィルターに対する許可を表示するには、Data Lake 管理者であるか、データフィルターに対する必要な許可を持っている必要があります。

Console

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/lakeformation/ で Lake Formation コンソールを開きます。
2. ナビゲーションペインの [Permissions] (許可) で [Data permissions] (データの許可) を選択します。
3. [Data permissions] (データの許可) ページで検索フィールドをクリックまたはタップし、[Properties] (プロパティ) メニューで [Resource type] (リソースタイプ) を選択します。
4. [Resource type] (リソースタイプ) メニューで [Resource type: Data cell filter] (リソースタイプ: データセルフィルター) を選択します。

   許可を持っているデータフィルターがリストされます。[Permissions] (許可) と [Grantable] (付与可能) 列を見るには、水平方向にスクロールする必要がある場合があります。



AWS CLI

- list-permissions のコマンドを入力します。resource 引数に DataCellsFilter を指定し、Permissions 引数、およびオプションで PermissionsWithGrantOption 引数に、DESCRIBE または DROP を指定します。

  以下の例は、データフィルター restrict-pharma に対する grant オプション付きの DESCRIBE 許可をリストします。結果は、 AWS アカウント 1111-2222-3333 のsalesデータベース内のプリンシパルdatalake_user1とordersテーブルに付与されたアクセス許可に限定されます。

```
aws lakeformation list-permissions --cli-input-json file://list-params.json
```

  以下は、ファイル grant-params.json の内容です。

```
{
    "Principal": {"DataLakePrincipalIdentifier":
 "arn:aws:iam::111122223333:user/datalake_user1"},
    "Resource": {
        "DataCellsFilter": {
            "TableCatalogId": "111122223333",
            "DatabaseName": "sales",
            "TableName": "orders",
            "Name": "restrict-pharma"
        }
    },
    "Permissions": ["DESCRIBE"],
    "PermissionsWithGrantOption": ["DESCRIBE"]
}
```

Lake Formation でのデータベースとテーブル許可の表示

Data Catalog データベースまたはテーブルについて付与された Lake Formation 許可を表示することができます。これを行うには、Lake Formation コンソール、 API、または AWS Command Line Interface () を使用しますAWS CLI。

コンソールを使用した許可の表示は、[Databases] (データベース) もしくは [Tables] (テーブル) ページ、または [Data permissions] (データの許可) ページから開始することができます。

> **Note**
>
> データベース管理者またはリソース所有者ではないときは、リソースに対する grant オプション付きの Lake Formation 許可がある場合に限り、他のプリンシパルが持っているそのリソースに対する許可を表示できます。
> 必要な Lake Formation アクセス許可に加えて、 AWS Identity and Access Management (IAM) アクセス許可 `glue:GetDatabases`、`glue:GetDatabase`、`glue:GetTables`、`glue:GetTable`、および が必要です`lakeformation:ListPermissions`。

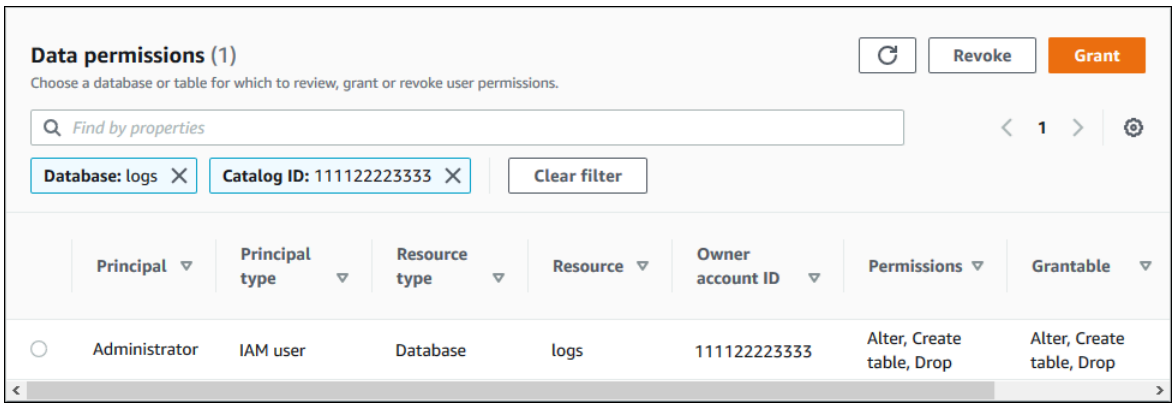
データベースに対する許可を表示する (コンソール。[Databases] (データベース) ページから開始)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、データベース作成者、またはデータベースに対する grant オプション付きの Lake Formation 許可を持つユーザーとしてサインインします。
2. ナビゲーションペインで、[Databases] (データベース) を選択します。
3. データベースを選択し、[Actions] (アクション) メニューで [View permissions] (許可を表示) を選択します。

   > **Note**
   >
   > データベースリソースリンクを選択する場合、Lake Formation はリソースリンクのターゲットデータベースではなく、リソースリンクに対する許可を表示します。

   [Data permissions] (データの許可) ページに、データベースに対するすべての Lake Formation 許可がリストされます。データベース所有者のデータベース名とカタログ ID (AWS アカウント ID) は、検索ボックスの下にラベルとして表示されます。タイルは、そのデータベースに対する許可のみをリストするようにフィルターが適用されたことを示します。タイルを閉じる、または [Clear filter] (フィルターをクリア) を選択することで、フィルターを調整することができます。



データベースに対する許可を表示する (コンソール。[Data permissions] (データの許可) ページから開始)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、データベース作成者、またはデータベースに対する grant オプション付きの Lake Formation 許可を持つユーザーとしてサインインします。
2. ナビゲーションペインで、[Data permissions] (データの許可) を選択します。
3. 検索ボックスにカーソルを置き、表示されるプロパティメニューでデータベースを選択します。
4. 表示される [Databases] (データベース) メニューで、データベースを選択します。

   > **Note**
   >
   > データベースリソースリンクを選択する場合、Lake Formation はリソースリンクのターゲットデータベースではなく、リソースリンクに対する許可を表示します。

   [Data permissions] (データの許可) ページに、データベースに対するすべての Lake Formation 許可がリストされます。データベース名が、検索ボックスの下にタイルとして表示されます。タイルは、そのデータベースに対する許可のみをリストするようにフィルターが適用されたことを示します。タイルを閉じる、または [Clear filter] (フィルターをクリア) を選択することで、フィルターを削除することができます。

テーブルに対する許可を表示する (コンソール。[Tables] (テーブル) ページから開始)

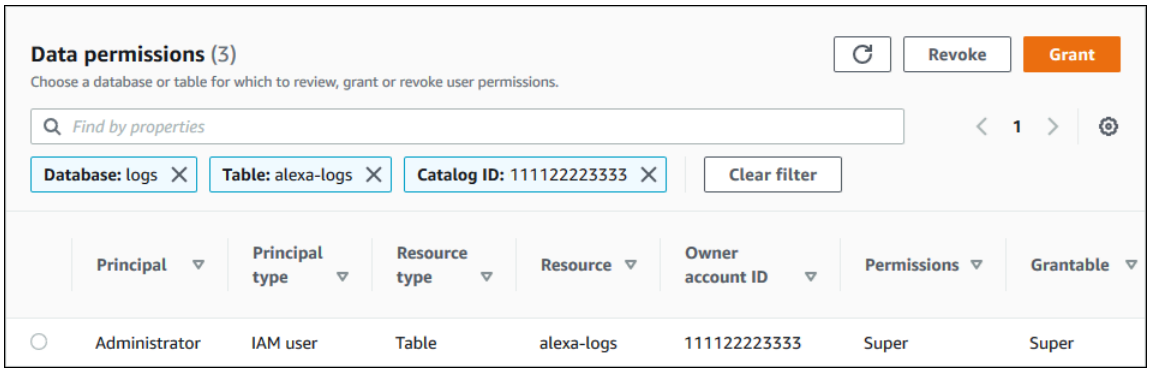
1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

データレイク管理者、テーブル作成者、またはテーブルに対する grant オプション付きの Lake Formation 許可を持つユーザーとしてサインインします。

2. ナビゲーションペインで [Table] (テーブル) を選択します。
3. テーブルを選択し、[Actions] (アクション) メニューで [View permissions] (許可を表示) を選択します。

> **Note**
>
> テーブルリソースリンクを選択する場合、Lake Formation はリソースリンクのターゲットテーブルではなく、リソースリンクに対する許可を表示します。

[Data permissions] (データの許可) ページに、テーブルに対するすべての Lake Formation 許可がリストされます。テーブル名、テーブルを含むデータベースのデータベース名、およびテーブル所有者のカタログ ID (AWS アカウント ID) は、検索ボックスの下にラベルとして表示されます。ラベルは、そのテーブルに対する許可のみをリストするようにフィルターが適用されたことを示します。ラベルを閉じる、または [Clear filter] (フィルターをクリア) を選択することで、フィルターを調整することができます。



テーブルに対する許可を表示する (コンソール。[Data permissions] (データの許可) ページから開始)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、テーブル作成者、またはテーブルに対する grant オプション付きの Lake Formation 許可を持つユーザーとしてサインインします。
2. ナビゲーションペインで、[Data permissions] (データの許可) を選択します。
3. 検索ボックスにカーソルを置き、表示されるプロパティメニューでデータベースを選択します。

4. 表示される [Databases] (データベース) メニューで、データベースを選択します。

   > **Important**
   >
   > 外部アカウントから AWS アカウントと共有されたテーブルに対するアクセス許可を表示するには、データベースへのリソースリンクではなく、テーブルを含む外部アカウントのデータベースを選択する必要があります。

   [Data permissions] (データの許可) ページに、データベースに対するすべての Lake Formation 許可がリストされます。

5. もう1 度検索ボックスにカーソルを置き、表示される [Properties] (プロパティ) メニューで [Table] (テーブル) を選択します。

6. 表示された [Tables] (テーブル) メニューで、テーブルを選択します。

   [Data permissions] (データの許可) ページに、テーブルに対するすべての Lake Formation 許可がリストされます。テーブル名と、テーブルが含まれるデータベースのデータベース名が、検索ボックスの下にタイルとして表示されます。タイルは、そのテーブルに対する許可のみをリストするようにフィルターが適用されたことを示します。タイルを閉じる、または [Clear filter] (フィルターをクリア) を選択することで、フィルターを調整することができます。

テーブルに対する許可を表示する (AWS CLI)

- `list-permissions` コマンドを入力します。

  以下の例は、外部アカウントから共有されているテーブルに対する許可をリストします。`CatalogId` プロパティは外部 AWS アカウントのアカウント ID であり、データベース名はテーブルを含む外部アカウントのデータベースを指します。

```
aws lakeformation list-permissions  --resource-type TABLE --resource '{ "Table":
 {"DatabaseName":"logs", "Name":"alexa-logs", "CatalogId":"123456789012"}}'
```

Lake Formation コンソールを使用した許可の取り消し

コンソールを使用して、Data Catalog 許可、ポリシータグ許可、データフィルター許可、およびロケーション許可といった、すべてのタイプの Lake Formation 許可を取り消すことができます。

リソースに対する Lake Formation 許可を取り消す (コンソール)

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、またはリソースに対する grant オプション付きの許可を付与されたユーザーとしてサインインします。
2. ナビゲーションペインの [許可] で、[データレイクのアクセス許可]、[LF タグとアクセス許可]、または [データのロケーション] を選択します。
3. 許可またはロケーションを選択してから、[Revoke] (取り消す) を選択します。
4. 表示されるダイアログボックスで、[Revoke] (取り消す) を選択します。

Lake Formation でのクロスアカウントデータ共有

Lake Formation のクロスアカウント機能を使用すると、ユーザーは分散データレイクを複数の AWS 組織間で安全に共有したり AWS アカウント、別のアカウントの IAM プリンシパルと直接共有したりして、データカタログのメタデータと基盤となるデータにきめ細かなアクセスを提供したりできます。大企業は通常、複数の を使用します。これらのアカウントの多くは AWS アカウント、単一のによって管理されるデータレイクにアクセスする必要がある場合があります AWS アカウント。ユーザーおよび AWS Glue 抽出、変換、ロード (ETL) ジョブは、複数のアカウント間でテーブルをクエリおよび結合できますが、Lake Formation のテーブルレベルおよび列レベルのデータ保護を活用できます。

Data Catalog リソースに対する Lake Formation アクセス許可を外部アカウントまたは別のアカウントの IAM プリンシパルに直接付与すると、Lake Formation は AWS Resource Access Manager (AWS RAM) サービスを使用してリソースを共有します。付与対象アカウントが付与する側のアカウントと同じ組織内にある場合、付与対象アカウントはその共有リソースをただちに使用できるようになります。被付与者アカウントが同じ組織にない場合、 は被付与者アカウントに招待 AWS RAM を送信して、リソース付与を承認または拒否します。次に、共有リソースを使用できるようにするには、被付与者アカウントのデータレイク管理者が AWS RAM コンソールまたは AWS CLI を使用して招待を受け入れる必要があります。

Lake Formation は、ハイブリッドアクセスモードでの外部アカウントとの Data Catalog リソースの共有をサポートしています。ハイブリッドアクセスモードでは、 AWS Glue Data Catalog内のデータベースとテーブルの Lake Formation 許可を柔軟かつ選択的に有効にできます。
ハイブリッドアクセスモードでは、他の既存のユーザーやワークロードのアクセス許可ポリシーを中断することなく、特定のユーザーのセットに Lake Formation 許可を設定できる増分パスが導入されました。

詳細については、「ハイブリッドアクセスモード」を参照してください。

直接的なクロスアカウント共有

許可されたプリンシパルは、外部アカウントの IAM プリンシパルとリソースを明示的に共有できます。この機能は、外部アカウントの誰がリソースにアクセスできるかをアカウント所有者が制御する場合に便利です。IAM プリンシパルが受け取るアクセス許可は、直接の付与とアカウントレベルの付与を組み合わせたもので、それらはプリンシパルにカスケードされます。アクセス許可の受領者のみが、直接クロスアカウント付与を表示できます。リソース共有を受け取るプリンシパルが、他のプリンシパルとリソースを共有することはできません。

データカタログリソースを共有する方法

単一の Lake Formation 付与操作で、以下の Data Catalog リソースに対するクロスアカウント許可を付与できます。

- 1 つのデータベース
- 個々のテーブル (オプションで列フィルタリングを使用)
- 選択された数個のテーブル
- データベース内のすべてのテーブル (すべてのテーブルのワイルドカードを使用)

データベースとテーブルを別の アカウントの別の AWS アカウント または IAM プリンシパルと共有するには、2 つのオプションがあります。

- Lake Formation のタグベースのアクセスコントロール (LF-TBAC) (推奨)

  Lake Formation のタグベースのアクセスコントロールは、属性に基づいて許可を定義する認可戦略です。タグベースのアクセスコントロールを使用して、Data Catalog リソース (データベース、テーブル、列) を外部の IAM プリンシパル、組織 AWS アカウント、組織単位 (OUs) と共有できます。これらの属性は、Lake Formation で LF タグと呼ばれています。詳細については、「Lake Formation のタグベースのアクセスコントロールを使用したデータレイクの管理」を参照してください。

  > **Note**
  >
  > クロスアカウント付与 AWS Resource Access Manager に使用する Data Catalog アクセス許可を付与する LF-TBAC メソッド。
  > Lake Formation では、LF-TBAC 方式を使用した Organizations および組織単位へのクロスアカウントアクセス許可の付与をサポートするようになりました。

この機能を有効にするには、[クロスアカウントのバージョン設定] を [バージョン 3] に更新する必要があります。
詳細については、「クロスアカウントデータ共有のバージョン設定の更新」を参照してください。

- Lake Formation の名前付きリソース

  名前付きリソース方式を使用した Lake Formation のクロスアカウントデータ共有を使用すると、外部、IAM プリンシパル AWS アカウント、組織、または組織単位に対して、Data Catalog テーブルとデータベースに対する許可オプションを使用して Lake Formation 許可を付与できます。この付与操作は、これらのリソースを自動的に共有します。

> **Note**
>
> Lake Formation 認証情報を使用して、 AWS Glue クローラが別のアカウントのデータストアにアクセスすることを許可することもできます。詳細については、「 AWS Glue デベロッパーガイド」の「クロスアカウントクローリング」を参照してください。

Athena や Amazon Redshift Spectrum などの統合されたサービスでは、クエリに共有リソースを含めることができるように、リソースリンクが必要になります。リソースリンクの詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。

考慮事項と制限事項については、「クロスアカウントデータ共有のベストプラクティスと考慮事項」を参照してください。

トピック

- 前提条件
- クロスアカウントデータ共有のバージョン設定の更新
- 外部アカウントから AWS アカウント または IAM プリンシパル間で Data Catalog テーブルとデータベースを共有する
- アカウントと共有されたデータベースまたはテーブルに対する許可の付与
- リソースリンク許可の付与
- 共有テーブルの基盤となるデータへのアクセス
- CloudTrail のクロスアカウントロギング
- AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理

- GetResourceShares API 操作を使用したすべてのクロスアカウント付与の表示

関連トピック

- Lake Formation 許可の概要
- 共有 Data Catalog テーブルとデータベースへのアクセスと表示
- リソースリンクの作成
- クロスアカウントアクセスのトラブルシューティング

前提条件

AWS アカウントが Data Catalog リソース (カタログ、データベース、テーブル) を別のアカウントまたは別のアカウントのプリンシパルと共有する前に、およびアカウントと共有されているリソースにアクセスする前に、次の前提条件を満たす必要があります。

クロスアカウントデータ共有の一般的な要件

- ハイブリッドアクセスモードでデータカタログのデータベースとテーブルを共有し、フェデレーティッドカタログでオブジェクトを共有するには、[クロスアカウントバージョン設定] を [バージョン 4] に更新する必要があります。
- Data Catalog リソースに対するクロスアカウント許可を付与する前に、そのリソースの `IAMAllowedPrincipals` グループからすべての Lake Formation 許可を取り消す必要があります。呼び出し元のプリンシパルがリソースにアクセスするためのクロスアカウント許可を持っていて、リソースに `IAMAllowedPrincipals` 許可がある場合、Lake Formation は `AccessDeniedException` をスローします。

  この要件は、基盤となるデータロケーションを Lake Formation モードで登録する場合にのみ該当します。データロケーションをハイブリッドモードで登録すると、`IAMAllowedPrincipals` グループ許可が共有データベースまたはテーブルに存在することになる可能性があります。
- 共有する予定のテーブルが含まれるデータベースについては、新しいテーブルに `IAMAllowedPrincipals` への `Super` のデフォルト付与がないようにする必要があります。Lake Formation コンソールで、データベースを編集してオフにします。このデータベースの新しいテーブルには IAM アクセスコントロールのみを使用するか、次の AWS CLI コマンドを入力して、 をデータベースの名前`database`に置き換えます。基になるデータロケーションがハイブリッドアクセスモードで登録されている場合は、このデフォルト設定を変更する必要はありません。ハイブ

リッドアクセスモードでは、Lake Formation ではAmazon S3と IAM アクセス許可ポリシーを同じリソース AWS Glue に選択的に適用できます。

```
aws glue update-database --name database --database-input
 '{"Name":"database","CreateTableDefaultPermissions":[]}'
```

- クロスアカウントアクセス許可を付与するには、付与者に AWS Glueおよび AWS RAM サービスに対する必要な AWS Identity and Access Management (IAM) アクセス許可が必要です。 AWS 管理ポリシーは、必要なアクセス許可AWSLakeFormationCrossAccountManagerを付与します。

  を使用してリソース共有を受信するアカウントのデータレイク管理者には、次の追加ポリシー AWS RAM が必要です。これにより、管理者は AWS RAM リソース共有の招待を受け入れることができます。また、管理者が組織とのリソース共有を有効にすることも可能にします。

  JSON

  ```
  {
      "Version":"2012-10-17",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "ram:AcceptResourceShareInvitation",
                  "ram:RejectResourceShareInvitation",
                  "ec2:DescribeAvailabilityZones",
                  "ram:EnableSharingWithAwsOrganization"
              ],
              "Resource": "*"
          }
      ]
  }
  ```

- Data Catalog リソースを AWS Organizations または組織単位と共有する場合は、 で組織との共有を有効にする必要があります AWS RAM。

  組織との共有を有効にする方法については、AWS RAM 「 ユーザーガイド」のAWS 「組織との共有を有効にする」を参照してください。

  組織との共有を有効にするには、ram:EnableSharingWithAwsOrganization 許可が必要です。

- 別のアカウントの IAM プリンシパルとリソースを直接共有するには、[Cross account version settings] (クロスアカウントバージョン設定) を [Version 3] (バージョン 3) に更新する必要があります。この設定は、[Data catalog settings] (データカタログ設定) ページにあります。[Version 1] (バージョン 1) を使用している場合は、設定を更新する手順「クロスアカウントデータ共有のバージョン設定の更新」を参照してください。
- AWS Glue サービスマネージドキーで暗号化された Data Catalog リソースを別のアカウントと共有することはできません。共有できるのは、お客様の暗号化キーで暗号化された Data Catalog リソースのみです。リソース共有を受け取るアカウントには、オブジェクトを復号するための Data Catalog 暗号化キーに対する許可が必要です。

LF-TBAC 要件を使用したクロスアカウントデータ共有

- Data Catalog リソースを AWS Organizations および組織単位 (OUs) と共有するには、クロスアカウントバージョン設定をバージョン 3 以上に更新する必要があります。
- Data Catalog リソースをバージョン 3 のクロスアカウントバージョン設定と共有するには、付与者はアカウントの AWS 管理ポリシー `AWSLakeFormationCrossAccountManager` で定義されている IAM アクセス許可を持っている必要があります。
- [クロスアカウントのバージョン設定] のバージョン 1 またはバージョン 2 を使用している場合、LF-TBAC を有効にする Data Catalog リソースポリシー (`glue:PutResourcePolicy`) が必要です。詳細については、「AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理」を参照してください。
- 現在 AWS Glue Data Catalog リソースポリシーを使用してリソースを共有しており、バージョン 3 以降のクロスアカウントバージョン設定を使用してクロスアカウントアクセス許可を付与する場合は、AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理「」セクションに示すように、 `glue:PutResourcePolicy` API オペレーションを使用して Data Catalog 設定に `glue:ShareResource` アクセス許可を追加する必要があります。AWS Glue Data Catalog リソースポリシー (バージョン 1 とバージョン 2 では `glue:PutResourcePolicy` の許可を使用) を使用してクロスアカウントアクセス付与を行わなかった場合、このポリシーは必要ありません。

```
{
      "Effect": "Allow",
      "Action": [
        "glue:ShareResource"
      ],
      "Principal": {"Service": [
        "ram.amazonaws.com"
      ]},
```

```
    "Resource": [
      "arn:aws:glue:<region>:<account-id>:table/*/*",
      "arn:aws:glue:<region>:<account-id>:database/*",
      "arn:aws:glue:<region>:<account-id>:catalog"
    ]
  }
```

- アカウントが AWS Glue データカタログリソースポリシーを使用してクロスアカウント共有を行っており、現在 AWS RAM を使用してリソースを共有するために名前付きリソースメソッドまたは LF-TBAC [クロスアカウント設定] バージョン 3 を使用してリソースを共有している場合は、`glue:PutResourcePolicy` API オペレーションを呼び出すときに引数 `EnableHybrid` を `'true'` に設定する必要があります。詳細については、「AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理」を参照してください。

共有リソースにアクセスする各アカウントで必要になるセットアップ

- リソースを共有する場合 AWS アカウント、共有リソースを表示するには、コンシューマーアカウントの少なくとも 1 人のユーザーがデータレイク管理者である必要があります。データレイク管理者の作成方法については、「データレイク管理者を作成する」を参照してください。

  データレイク管理者は、共有リソースに対する Lake Formation 許可をアカウント内の他のプリンシパルに付与できます。他のプリンシパルは、データレイク管理者から共有リソースに対する許可を付与されるまで、そのリソースにアクセスできません。
- Athena や Redshift Spectrum などの統合されたサービスでは、クエリに共有リソースを含めることができるように、リソースリンクが必要になります。プリンシパルは、その Data Catalog に、別の AWS アカウントアカウントからの共有リソースへのリソースリンクを作成する必要があります。リソースリンクの詳細については、「Lake Formation でのリソースリンクの仕組み」を参照してください。
- リソースを IAM プリンシパルと直接共有する場合、Athena を使用してテーブルをクエリする場合、プリンシパルはそのリソースリンクを作成する必要があります。リソースリンクを作成するには、プリンシパルは Lake Formation の `CREATE_TABLE` または `CREATE_DATABASE` アクセス許可と、`glue:CreateTable` または `glue:CreateDatabase` IAM アクセス許可が必要です。

  プロデューサーアカウントが同じデータベース内の別のテーブルを同じプリンシパルまたは別のプリンシパルと共有している場合、そのプリンシパルはすぐにテーブルをクエリできます。

> **Note**
>
> データレイク管理者と、データレイク管理者から許可が付与されたプリンシパルには、共有リソースがローカル (所有) リソースであるかのように Data Catalog に表示されます。抽出、変換、ロード (ETL) ジョブは、共有リソースの基盤となるデータにアクセスできます。共有リソースについては、Lake Formation コンソールの [Tables] (テーブル) および [Databases] (データベース) ページに所有者のアカウント ID が表示されます。
> 共有リソースの基盤となるデータに対するアクセスが行われると、共有リソース受領者のアカウントと、リソース所有者のアカウントの両方で CloudTrail ログイベントが生成されます。CloudTrail イベントには、データにアクセスしたプリンシパルの ARN を含めることができますが、これは受領者アカウントがログにプリンシパル ARN を含めるようにオプトインする場合のみになります。詳細については、「CloudTrail のクロスアカウントロギング」を参照してください。

クロスアカウントデータ共有のバージョン設定の更新

は、クロスアカウントデータ共有設定を随時 AWS Lake Formation 更新して、 AWS RAM 使用量に加えられた変更を区別し、クロスアカウントデータ共有機能に加えられた更新をサポートします。Lake Formation がこれを行うと、[Cross account version settings] (クロスアカウントバージョン設定) の新しいバージョンが作成されます。

クロスアカウントバージョン設定の主な違い

さまざまな [Cross account version settings] (クロスアカウントバージョン設定) でのクロスアカウントデータ共有の仕組みの詳細については、以下のセクションを参照してください。

> **Note**
>
> 別のアカウントとデータを共有するには、付与者が `AWSLakeFormationCrossAccountManager` マネージド IAM ポリシーのアクセス許可を持っている必要があります。これがすべてのバージョン必須の前提条件です。
> [Cross account version settings] (クロスアカウントバージョン設定) を更新しても、共有リソースに対する受信者のアクセス許可には影響しません。これは、バージョン 1 からバージョン 2、バージョン 2 からバージョン 3、バージョン 1 からバージョン 3 への更新の場合に適用されます。バージョンを更新する際は、以下の考慮事項を参照してください。

バージョン 1

名前付きリソースメソッド: 各クロスアカウント Lake Formation アクセス許可付与を 1 つの AWS RAM リソース共有にマッピングします。ユーザー (付与者ロールまたはプリンシパル) には追加のアクセス許可は必要ありません。

LF-TBAC メソッド: クロスアカウント Lake Formation アクセス許可の付与は、データの共有 AWS RAM に を使用しません。ユーザーには `glue:PutResourcePolicy` アクセス許可が必要です。

バージョン更新のメリット: 初期バージョン - 該当しません

バージョンを更新する際の考慮事項: 初期バージョン - 該当しません

バージョン 2

名前付きリソースメソッド: 複数のクロスアカウントアクセス許可の付与を 1 つの AWS RAM リソース共有にマッピングすることで、 AWS RAM リソース共有の数を最適化します。ユーザーには、追加のアクセス許可は必要ありません。

LF-TBAC メソッド: クロスアカウント Lake Formation アクセス許可の付与は、データの共有 AWS RAM に を使用しません。ユーザーには `glue:PutResourcePolicy` アクセス許可が必要です。

バージョンを更新するメリット: AWS RAM 容量の最適な使用率によるスケーラブルなクロスアカウント設定。

バージョンを更新する際の考慮事項: クロスアカウント Lake Formation アクセス許可を付与するユーザーには、 `AWSLakeFormationCrossAccountManager` AWS マネージドポリシーの アクセス許可が必要です。それ以外の場合は、別のアカウントとリソースを正常に共有するための `ram:AssociateResourceShare` および `ram:DisassociateResourceShare` アクセス許可が必要です。

バージョン 3

名前付きリソースメソッド: 複数のクロスアカウントアクセス許可の付与を 1 つの AWS RAM リソース共有にマッピングすることで、 AWS RAM リソース共有の数を最適化します。ユーザーには、追加のアクセス許可は必要ありません。

LF-TBAC メソッド: Lake Formation はクロスアカウント付与 AWS RAM に を使用します。ユーザーは `glue:PutResourcePolicy` アクセス許可に glue: ShareResource ステートメントを

追加する必要があります。受信者は、 からのリソース共有の招待を受け入れる必要があります AWS RAM。

バージョン更新のメリット: 次の機能をサポートします。

- 外部アカウントの IAM プリンシパルとリソースを明示的に共有できます。

  詳細については、「データカタログリソースに対するアクセス許可の付与」を参照してください。
- Organizations または組織単位 (OU) に対して、LF-TBAC 方式を使用したクロスアカウント共有を可能にします。
- クロスアカウント付与の追加 AWS Glue ポリシーを維持するオーバーヘッドを排除します。

バージョンを更新する際の考慮事項: LF-TBAC 方式を使用してリソースを共有する場合、付与者がバージョン 3 より前のバージョンを使用していて、受信者がバージョン 3 以降を使用していると、付与者に「Invalid cross account grant request. Consumer account has opt-in to cross account version: v3. Please update `CrossAccountVersion` in `DataLakeSetting` to minimal version v3 (Service: AmazonDataCatalog; Status Code: 400; Error Code: InvalidInputException)」というエラーメッセージが表示されます。ただし、付与者がバージョン 3 を使用していて、受信者がバージョン 1 またはバージョン 2 を使用している場合、LF タグを使用したクロスアカウント付与は正常に行われます。

名前付きリソース方式を使用したクロスアカウント付与は、異なるバージョン間で互換性があります。付与者アカウントが以前のバージョン (バージョン 1 または 2) を使用していて、受信者アカウントが新しいバージョン (バージョン 3 以降) を使用している場合でも、クロスアカウントアクセス機能は互換性の問題やエラーなしでシームレスに動作します。

リソースを別のアカウントの IAM プリンシパルと直接共有するには、付与者だけがバージョン 3 を使用する必要があります。

LF-TBAC 方式を使用してクロスアカウント付与を行うには、ユーザーがアカウントに AWS Glue Data Catalog リソースポリシーを持っている必要があります。バージョン 3 に更新すると、LF-TBAC は AWS RAMを使用して付与します。 AWS RAM ベースのクロスアカウント許可を成功させるには、 AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理セクションに示すように、 `glue:ShareResource`ステートメントを既存の Data Catalog リソースポリシーに追加する必要があります。

バージョン 4

ハイブリッドアクセスモードでデータカタログリソースを共有したり、フェデレーティッドカタログ内のオブジェクトを共有したりするには、付与者にバージョン 4 以降が必要です。

バージョン 5

クロスアカウントバージョン 5 は、クロスアカウントリソース共有を強化し、無制限の数のテーブルを別のアカウントと共有できるようにします。これにより、リソースタイプごとの以前のリソース関連付け制限がなくなります。開始するには、Lake Formation コンソールまたは API を使用してクロスアカウントバージョン 5 にアップグレードします。新しいクロスアカウントアクセス許可の付与では、個々のリソースの関連付けではなく、リソース共有でワイルドカードパターンが自動的に使用されます。既存のクロスアカウント共有はすべて引き続き機能し、既存の Lake Formation APIs はすべて互換性を維持します。

バージョンを更新するメリット: クロスアカウント v5 はクロスアカウント共有を強化し、アカウント間で数十万のテーブルを共有できます。

バージョンを更新する際の考慮事項: バージョン 5 のアップグレード後の新しい権限では、既存の AWS Resource Manager リソース共有にワイルドカードリソースパターンが追加されるか、ワイルドカードパターンで新しい共有が作成されます。バージョン 5 にアップグレードすると、ダウングレードはサポートされません。

AWS RAM リソース共有の最適化

クロスアカウント付与の新しいバージョン (バージョン 2 以降) では、クロスアカウントの使用を最大化するために AWS RAM 容量が最適に活用されます。外部 AWS アカウント または IAM プリンシパルとリソースを共有すると、Lake Formation は新しいリソース共有を作成するか、リソースを既存の共有に関連付けることができます。Lake Formation は、既存の共有と関連付けることによって、コンシューマーが受け入れる必要があるリソース共有への招待数を減らします。バージョン 5 では、個々のリソースの関連付けの代わりにワイルドカードベースのリソースパターンを使用することで、RAM の使用がさらに最適化されるため、リソース共有あたりのリソースの関連付けが大幅に削減されます。

TBAC 経由で AWS RAM 共有を有効にするか、リソースをプリンシパルに直接共有する

リソースを別のアカウントの IAM プリンシパルと直接共有するか、Organizations や組織単位との TBAC クロスアカウント共有を有効にするには、[Cross account version settings] (クロスアカウントバージョン設定) を [Version 3] (バージョン 3) に更新する必要があります。 AWS RAM リソース制限の詳細については、「」を参照してください[クロスアカウントデータ共有のベストプラクティスと考慮事項](#)。

クロスアカウントのバージョン設定の更新に必要なアクセス許可

クロスアカウント許可の付与者に AWSLakeFormationCrossAccountManager マネージド IAM ポリシーのアクセス許可がある場合、クロスアカウントアクセス許可の付与者ロールまたはプリンシパルに追加のアクセス許可設定は必要ありません。ただし、クロスアカウントの付与者がマネージドポリシーを使用していない場合、新しいバージョンのクロスアカウント付与を成功させるには、付与者ロールまたはプリンシパルに次の IAM 許可が付与されている必要があります。

JSON

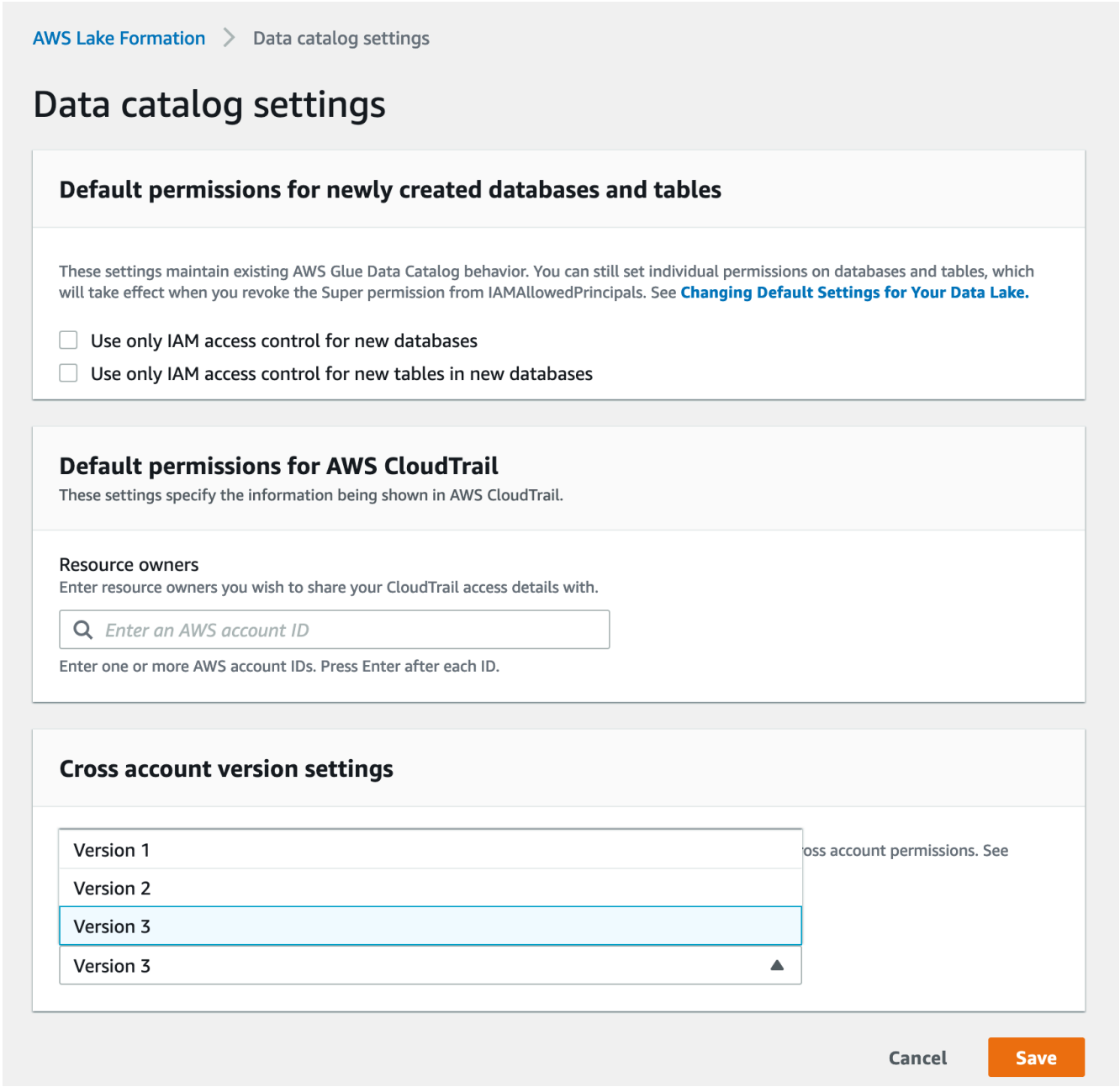
```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
         "ram:AssociateResourceShare",
         "ram:DisassociateResourceShare",
         "ram:GetResourceShares"
       ],
     "Resource": "*",
     "Condition": {
       "StringLike": {
         "ram:ResourceShareName": "LakeFormation*"
        }
       }
    }
  ]
}
```

新しいバージョンを有効にするには

コンソールまたは を使用してクロスアカウントバージョン設定を更新するには、 AWS Lake Formation次の手順に従います AWS CLI。

Console

1. データカタログ設定ページのクロスアカウントバージョン設定で、バージョン 2、バージョン 3、バージョン 4、またはバージョン 5 を選択します。[Version 1] (バージョン 1) を選択すると、Lake Formation はデフォルトのリソース共有モードを使用します。



2. [保存] を選択します。

AWS Command Line Interface (AWS CLI)

`put-data-lake-settings` AWS CLI コマンドを使用して `CROSS_ACCOUNT_VERSION`パラメータを設定します。使用できる値は 1、2、3、4、5 です。

```
aws lakeformation put-data-lake-settings --region us-east-1 --data-lake-settings
 file://settings
{
    "DataLakeAdmins": [
        {
            "DataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/test"
        }
    ],
    "CreateDatabaseDefaultPermissions": [],
    "CreateTableDefaultPermissions": [],
    "Parameters": {
        "CROSS_ACCOUNT_VERSION": "3"
    }
}
```

> **Important**
>
> [Version 2] (バージョン 2) または [Version 3] (バージョン 3) を選択すると、新しい名前付きリソースの付与はすべて新しいクロスアカウント付与モードになります。既存のクロスアカウント共有の AWS RAM 容量を最適に使用するには、古いバージョンで行われた許可を取り消し、新しいモードで再付与することをお勧めします。

外部アカウントから AWS アカウント または IAM プリンシパル間で Data Catalog テーブルとデータベースを共有する

このセクションでは、Data Catalog リソースに対するクロスアカウントアクセス許可を外部 AWS アカウント、IAM プリンシパル、 AWS 組織、または組織単位に付与する方法について説明します。この付与操作は、これらのリソースを自動的に共有します。

トピック

- タグベースのアクセスコントロールを使用したデータ共有
- 名前付きリソース方式を使用したクロスアカウントデータ共有。

タグベースのアクセスコントロールを使用したデータ共有

AWS Lake Formation タグベースのアクセスコントロール (LF-TBAC) は、属性に基づいてアクセス許可を定義する認可戦略です。以下の手順では、LF タグを使用してクロスアカウント許可を付与する方法を説明します。

プロデューサー/付与者アカウントで必要なセットアップ

1. LF タグを追加します。

   1. Lake Formation コンソールにデータレイク管理者または LF タグ作成者としてサインインします。
   2. ナビゲーションペインで、[アクセス許可] の [LF タグとアクセス許可] を選択します。
   3. [Add LF-Tag] (LF タグを追加) を選択します。

      LF タグの作成手順については、「LF タグの作成」を参照してください。

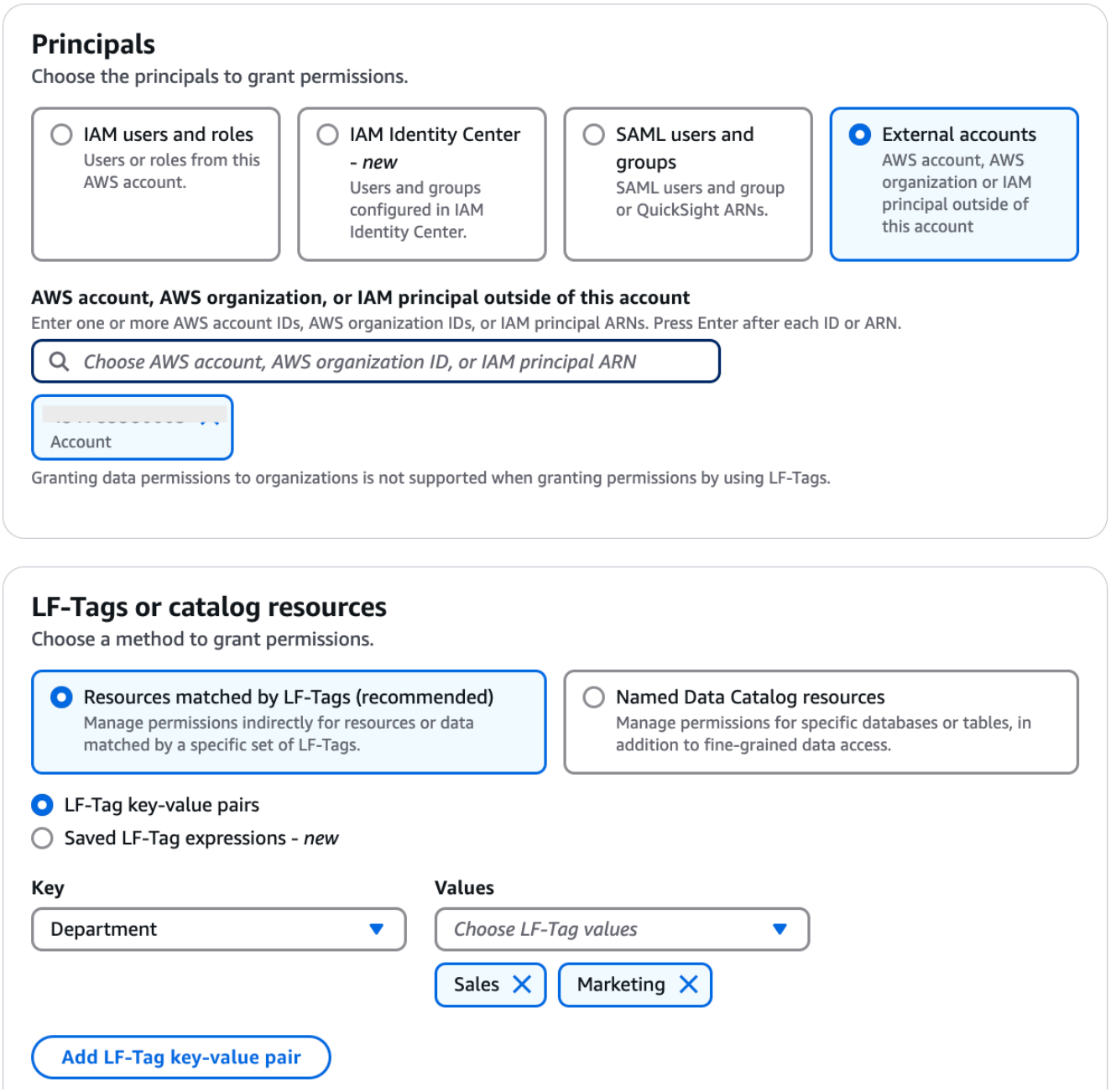
2. 自分のカウントまたは外部アカウントの IAM プリンシパルに、LF タグのキーと値のペアの記述および/または関連付けのアクセス許可を付与します。

   LF タグのキーと値のペアに対するアクセス許可を付与すると、プリンシパルは LF タグを表示し、それをデータカタログリソース (データベース、テーブル、列) に割り当てることができるようになります。

3. 次に、データレイク管理者または関連付けアクセス許可を持つ IAM プリンシパルは、データベース、テーブル、または列に LF タグを割り当てることができます。詳細については、「Data Catalog リソースへの LF タグの割り当て」を参照してください。

4. 次に、LF タグ式を使用して外部アカウントにデータアクセス許可を付与します。これにより、アクセス許可の被付与者または受領者は、同じキーと値でタグ付けされたデータカタログリソースにアクセスできます。

   1. ナビゲーションペインの [アクセス許可] で [データのアクセス許可] を選択します。
   2. [付与] を選択します。
   3. [アクセス許可の付与] ページで、[プリンシパル] に [外部アカウント] を選択し、外部プリンシパルに直接クロスアカウント付与を行う場合は、プリンシパルの被付与者 AWS アカウント ID もしくは IAM ロール、またはプリンシパルの Amazon リソースネーム (ARN) (プリンシパル ARN) を入力します。アカウント ID を入力した後、Enter キーを押します。



4. [LF タグまたはカタログリソース] で、[LF タグに一致するリソース (推奨)] を選択します。
   a. オプションの [LF タグのキーと値のペア] または [保存された LF タグ式] を選択します。
   b. [LF タグのキーと値のペア] を選択した場合は、被付与者アカウントと共有されているデータカタログリソースに関連付けられている LF タグのキーと値を入力します。

      被付与者には、LF タグ式で一致する LF タグが割り当てられたデータカタログリソースに対するアクセス許可が付与されます。LF タグ式がタグキーごとに複数の値を指定する場合、タグ値のいずれかを一致させることができます。

5. LF タグ式と一致するリソースに付与する、データベースレベルまたはテーブルレベルのアクセス許可を選択します。

> **Important**
>
> データレイク管理者は、共有リソースに対するアクセス許可を被付与者アカウント内のプリンシパルに付与する必要があるため、クロスアカウントアクセス許可は、常に付与オプションと共に付与される必要があります。

詳細については、「コンソールを使用した LF-Tag アクセス許可の付与」を参照してください。

> **Note**
>
> クロスアカウント付与を直接受け取るプリンシパルには、[Grantable permissions] (付与可能なアクセス許可) オプションがありません。

受信者側/被付与者アカウントで必要なセットアップ

1. Lake Formation コンソールにデータレイク管理者としてサインインします。
2. 次に、コンシューマーアカウントでリソース共有を受け取ります。
    1. AWS RAM コンソールを開きます。
    2. ナビゲーションペインの [自分と共有]で、[リソース共有] を選択します。
    3. リソース共有を選択し、[リソース共有を受け入れる] を選択します。
3. 別のアカウントとリソースを共有しても、そのリソースは引き続きプロデューサーアカウントに属し、Athena コンソール内には表示されません。リソースを Athena コンソールで表示するには、共有リソースを指すリソースリンクを作成する必要があります。リソースリンクの作成手順については、「共有 Data Catalog テーブルへのリソースリンクの作成」および「共有 Data Catalog データベースへのリソースリンクの作成」を参照してください。
    1. データカタログで [データベース] または [テーブル] を選択します。
    2. [データベース/テーブル] ページで、作成]、[リソースリンク] を選択します。
    3. データベースリソースリンクに次の情報を入力します。
        - リソースリンク名 – リソースリンクの一意の名前。

- 送信先カタログ – リソースリンクを作成するカタログ。
- 共有データベースのリージョン - 別のリージョンでリソースリンクを作成する場合は、共有するベースのリージョン。
- 共有データベース – 共有データベースを選択します。
- 共有データベースのカタログ ID – 共有データベースのカタログ ID を入力します。

4. [作成] を選択します。新しく作成されたリソースリンクは、データベースリストに表示されます。

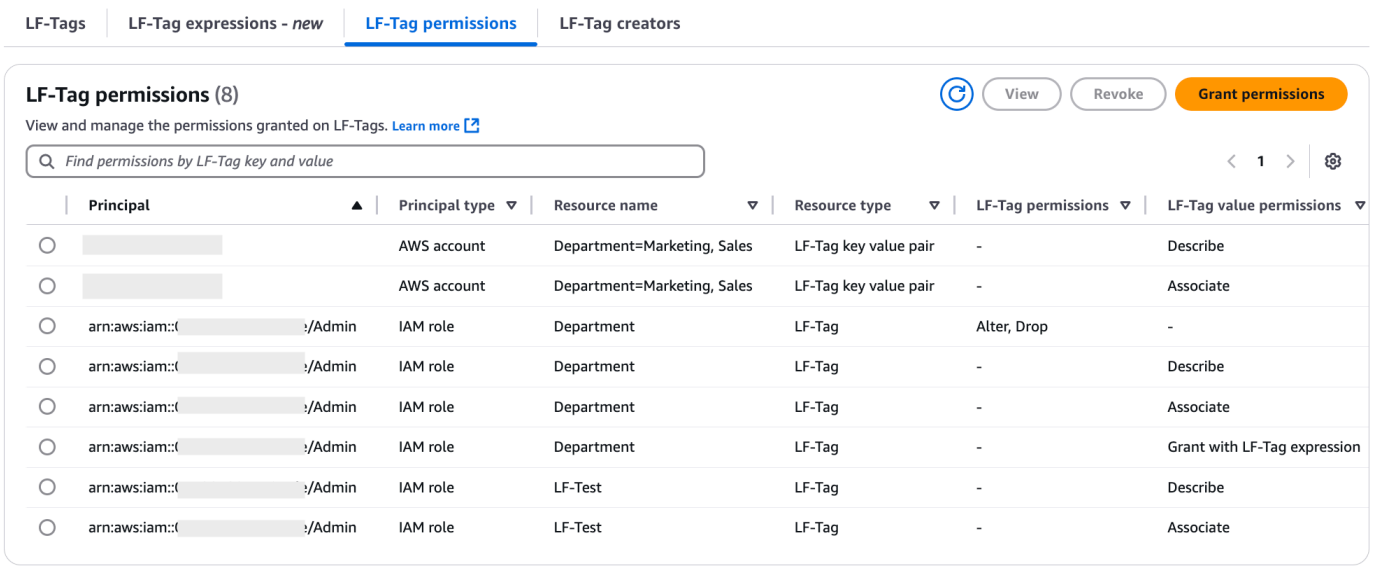
同様に、共有テーブルへのリソースリンクを作成できます。

4. ここで、リソースを共有する IAM プリンシパルへのリソースリンクに対する 記述アクセス許可を付与します。

   1. [データベース/テーブル]ページでリソースリンクを選択し、[アクション] メニューで、[付与] を選択します。
   2. [アクセス許可を付与] セクションで、[IAM ユーザーとロール] を選択します。
   3. リソースリンクへのアクセスを許可する IAM ロールを選択します。
   4. [リソースリンク] のアクセス許可セクションで、[記述] を選択します。
   5. [付与] を選択します。

5. 次に、コンシューマーアカウントのプリンシパルに [LF タグのキーと値のアクセス許可] を付与します。

   共有されている LF タグは、Lake Formation コンソールのコンシューマーアカウントにある、[アクセス許可] の [LF タグとアクセス許可] で確認できます。付与者から共有されたタグを、データベース、テーブル、列を含む付与者アカウントから共有されたリソースに関連付けることができます。また、リソースに対するアクセス許可を他のプリンシパルに付与できます。



1. ナビゲーションペインで、[アクセス許可]、[データレイクのアクセス許可] の順に移動し、[付与] を選択します。
2. [アクセス許可の付与] ページで、[IAM ユーザーとロール] を選択します。
3. 次に、アカウント内の IAM ユーザーとロールを選択して、共有データベース/テーブルへのアクセスを許可します。
4. [LF タグまたはカタログリソース] で、[LF タグに一致するリソース] を選択します。
5. 次に、共有されている LF タグのキーと値を選択します。
6. 次に、IAM ユーザーとロールに付与するデータベースとテーブルのアクセス許可を選択します。また、IAM ユーザーとロールが他のユーザー/ロールにアクセス許可を付与できるようにする [付与可能なアクセス許可] を選択することもできます。
7. [付与] を選択します。
8. Lake Formation コンソールの [データのアクセス許可] でアクセス許可の付与を表示できます。

名前付きリソース方式を使用したクロスアカウントデータ共有。

へのアクセス許可は、別の AWS アカウントのプリンシパル、または外部 AWS アカウント または に直接付与できます AWS Organizations。Lake Formation のアクセス許可を Organizations または組織単位に付与することは、その組織または組織単位 AWS アカウント のすべての にアクセス許可を付与することと同じです。

外部のアカウントまたは組織にアクセス許可を付与する場合は、[Grantable permissions] (付与可能なアクセス許可) オプションを含める必要があります。共有リソースにアクセスできるのは、外部アカウント内のデータレイク管理者が外部アカウント内の他のプリンシパルに共有リソースに対する許可を付与するまで、データレイク管理者のみになります。

> **Note**
>
> 外部アカウントから IAM プリンシパルに直接アクセス許可を付与する場合、[Grantable permissions] (付与可能なアクセス許可) オプションはサポートされません。

「名前付きリソース方式を使用したデータベースのアクセス許可の付与」の手順に従い、名前付きリソース方式を使用してクロスアカウント許可を付与します。

アカウントと共有されたデータベースまたはテーブルに対する許可の付与

別の AWS アカウントに属する Data Catalog リソースがアカウント AWS と共有されると、データレイク管理者として、共有リソースに対するアクセス許可をアカウントの他のプリンシパルに付与できます。ただし、リソースに対する許可を他の AWS アカウントまたは組織に付与することはできません。

AWS Lake Formation コンソール、API、または AWS Command Line Interface (AWS CLI) を使用してアクセス許可を付与できます。

共有データベースに対する許可を付与する (名前付きリソース方式、コンソール)

- 「名前付きリソース方式を使用したデータベースのアクセス許可の付与」の手順を実行します。[LF-Tags or catalog resources] (LF タグまたはカタログリソース) の [Database] (データベース) リストでは、外部アカウントのデータベースを選択して、データベースのリソースリンクは選択しないようにしてください。

  データベースのリストにデータベースが表示されない場合は、そのデータベースの AWS Resource Access Manager (AWS RAM) リソース共有招待を承諾していることを確認してください。詳細については、「からのリソース共有の招待の承諾 AWS RAM」を参照してください。

  また、CREATE_TABLE および ALTER 許可については、「データロケーション許可の付与 (同じアカウント)」の手順を実行し、[Registered account location] (登録されたアカウントのロケーション) に所有側のアカウント ID を入力するようにしてください。

共有テーブルに対する許可を付与する (名前付きリソース方式、コンソール)

- 「名前付きリソース方式を使用したテーブル許可の付与」の手順を実行します。[LF-Tags or catalog resources] (LF タグまたはカタログリソース) の [Database] (データベース) リストでは、外部アカウントのデータベースを選択して、データベースのリソースリンクは選択しないようにしてください。

  テーブルのリストにテーブルが表示されない場合は、そのテーブルの AWS RAM リソース共有招待を承諾していることを確認してください。詳細については、「からのリソース共有の招待の承諾 AWS RAM」を参照してください。

  また、ALTER 許可については、「データロケーション許可の付与 (同じアカウント)」の手順を実行し、[Registered account location] (登録されたアカウントのロケーション) に所有側のアカウント ID を入力するようにしてください。

共有リソースに対する許可を付与する (LF-TBAC 方式、コンソール)

- 「データカタログ許可の付与」の手順を実行します。[LF タグまたはカタログリソース] セクションで、外部アカウントがアカウントに付与したものと同一の LF タグ式、またはその式のサブセットを付与します。

  例えば、外部アカウントが LF タグ式 `module=customers AND environment=production` を付与オプションでアカウントに付与した場合は、データレイク管理者として、同じ式や、`module=customers` または `environment=production` をアカウント内のプリンシパルに付与できます。付与できるのは、リソースに対して LF タグ式で付与された Lake Formation 許可 (例えば `SELECT` や `ALTER` など) と同じ許可、またはそのサブセットのみです。

共有テーブルに対するアクセス許可を付与するには (名前付きリソースメソッド AWS CLI)

- 以下のようなコマンドを入力します。この例では、以下のようになっています。
  - AWS アカウント ID は 1111-2222-3333 です。
  - テーブルを所有し、それをアカウントに付与したアカウントは 1234-5678-9012 です。
  - 共有テーブル `pageviews` に対する `SELECT` 許可がユーザー `datalake_user1` に付与されています。そのユーザーはアカウントのプリンシパルです。
  - `pageviews` テーブルは、アカウント 1234-5678-9012 が所有する `analytics` データベースにあります。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "SELECT" --resource '{ "Table": {"CatalogId":"123456789012",
 "DatabaseName":"analytics", "Name":"pageviews"}}'
```

`resource` 引数の `CatalogId` プロパティには、所有側のアカウントを指定する必要があることに注意してください。

リソースリンク許可の付与

AWS アカウントのプリンシパルに 1 つ以上のリソースリンクに対する AWS Lake Formation アクセス許可を付与するには、次の手順に従います。

リソースリンクの作成後は、作成したユーザーのみがそのリンクを表示してアクセスすることができます。(これは、データベースに [Use only IAM access control for new tables in this database] (このデータベース内の新しいテーブルには IAM アクセスコントロールのみを使用する) が有効化されていないことを前提としています。) アカウント内の他のプリンシパルがリソースリンクにアクセスすることを許可するには、少なくとも `DESCRIBE` 許可を付与してください。

> **Important**
>
> リソースリンクに対する許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルに対する許可は付与されません。ターゲットに対する許可は、別途付与する必要があります。

Lake Formation コンソール、 API、または AWS Command Line Interface () を使用してアクセス許可を付与できますAWS CLI。

console

Lake Formation コンソールを使用してリソースリンク許可の付与するには

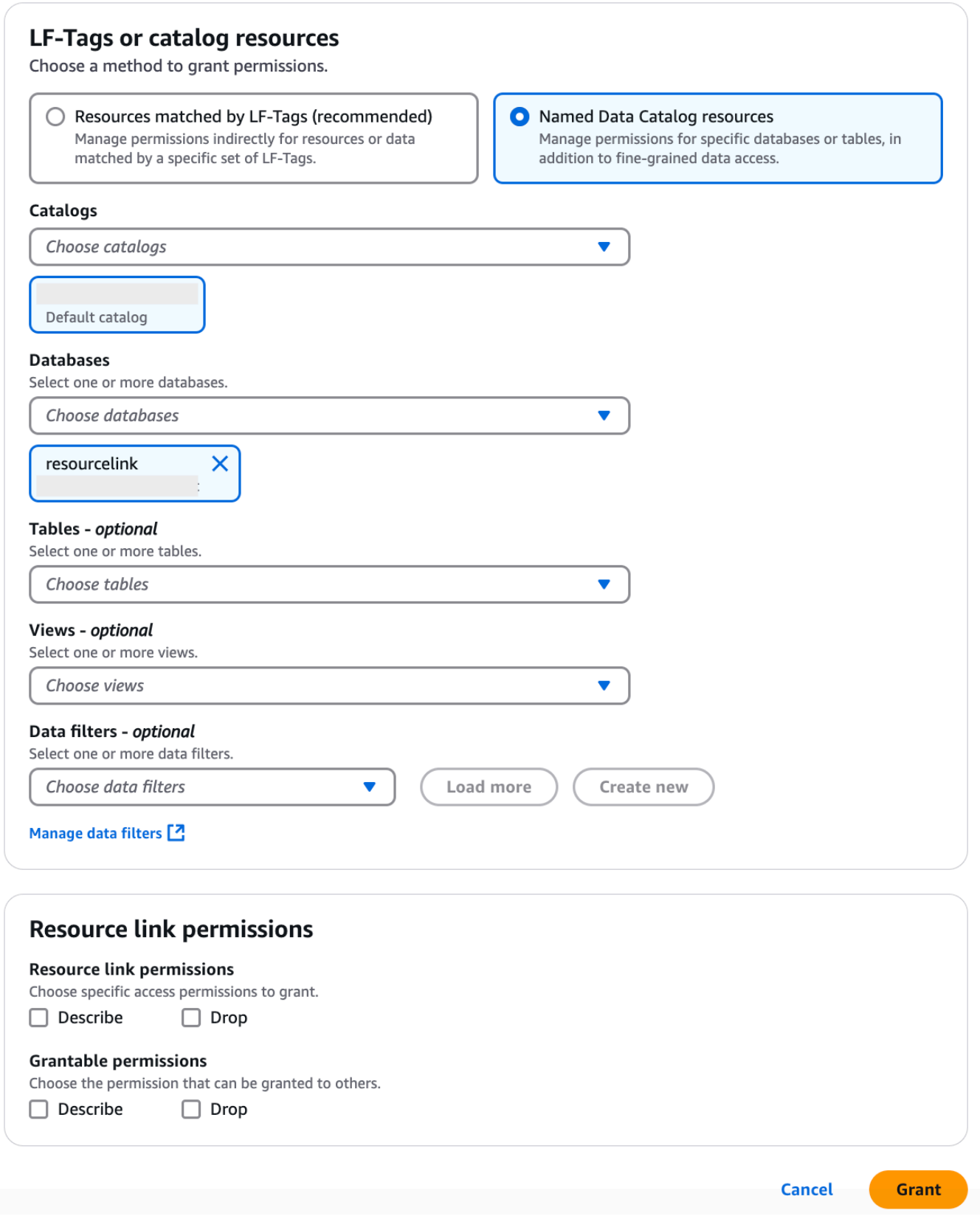
1. 次のいずれかを行います。

   - データベースリソースリンクの場合は、「名前付きリソース方式を使用したデータベースのアクセス許可の付与」の手順に従って以下を実行します。

1. [データカタログ] の [データベース] で、データベースリストからリソースリンクを選択します。
2. [付与] を選択して[アクセス許可の付与] ページを開きます。
3. アクセス許可を付与するプリンシパルを指定します。
4. [カタログ] フィールドと [データベース] フィールドにデータが入力されます。

- テーブルリソースリンクの場合は、「名前付きリソース方式を使用したテーブル許可の付与」の手順に従って以下を実行します。
  1. [データカタログ] の [テーブル] で、テーブルリストからリソースリンクを選択します。
  2. [アクセス許可の付与] ページを開きます。
  3. プリンシパルを指定します。
  4. [カタログ]、[データベース]、[テーブル] フィールドにデータが入力されます。
  5. プリンシパルを指定します。

2. [Permissions] (許可) で、付与する許可を選択します。オプションで、[Grantable Permissions] (付与可能な許可) を選択します。



3. [Grant] (付与) を選択します。

AWS CLI

を使用してリソースリンクのアクセス許可を付与するには AWS CLI

- リソースリンクをリソースとして指定して、grant-permissions コマンドを実行します。

  Example

  この例ではDESCRIBE、 AWS アカウント 1111-2222-3333 datalake_user1のデータベースincidents-link内のテーブルリソースリンクissuesの ユーザーに を付与します。

```
aws lakeformation grant-permissions --principal
 DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1
 --permissions "DESCRIBE" --resource '{ "Table": {"DatabaseName":"issues",
 "Name":"incidents-link"}}'
```

以下も参照してください。

- リソースリンクの作成
- Lake Formation 許可のリファレンス

共有テーブルの基盤となるデータへのアクセス

AWS アカウント A がデータカタログテーブルをアカウント B と共有しているとします。たとえば、テーブルの grant オプションSELECTを使用して をアカウント B に付与します。アカウント B のプリンシパルが共有テーブルの基盤となるデータを読み取れるようにするには、次の条件を満たす必要があります。

- アカウント B のデータレイク管理者が共有を承諾すること。(これは、アカウント A と B が同じ組織内にある場合、またはこの付与が Lake Formation のタグベースのアクセスコントロール方式で行われた場合は必要ありません。)
- アカウント A が付与した共有テーブルに対する Lake Formation SELECT 許可を、データレイク管理者がプリンシパルに再度付与すること。
- プリンシパルが、テーブル、テーブルが含まれるデータベース、およびアカウント A Data Catalog に対する以下の IAM 許可を持っていること。

> **Note**
>
> 以下の IAM ポリシーで、これらを実行してください。
>
> - *<account-id-A>* を AWS アカウント A のアカウント ID に置き換えます。
> - *<region>* を有効なリージョンに置き換える。
> - *<database>* を、アカウント A 内の共有テーブルが含まれるデータベースの名前に置き換える。
> - *<table>* を共有テーブルの名前に置き換える。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "glue:GetTable",
            "glue:GetTables",
            "glue:GetPartition",
            "glue:GetPartitions",
            "glue:BatchGetPartition",
            "glue:GetDatabase",
            "glue:GetDatabases"
           ],
           "Resource": [
            "arn:aws:glue:us-east-1:111122223333:table/<database>/<table>",
            "arn:aws:glue:us-east-1:111122223333:database/<database>",
            "arn:aws:glue:us-east-1:111122223333:catalog"
           ]
        },
        {
          "Effect": "Allow",
          "Action": [
            "lakeformation:GetDataAccess"
           ],
          "Resource": [
            "*"
```

```
            ]
        }
    ]
}
```

以下も参照してください。

- からのリソース共有の招待の承諾 AWS RAM

CloudTrail のクロスアカウントロギング

Lake Formation は、データレイク内のデータに対するすべてのクロスアカウントアクセスの一元的な監査証跡を提供します。受信者 AWS アカウントが共有テーブルのデータにアクセスすると、Lake Formation は CloudTrail イベントを所有アカウントの CloudTrail ログにコピーします。コピーされたイベントには、 Amazon Athena や Amazon Redshift Spectrum などの統合サービスによるデータに対するクエリと、AWS Glueジョブによるデータアクセスが含まれます。

Data Catalog リソースへのクロスアカウント操作に関する CloudTrail イベントも、同様にコピーされます。

リソース所有者として Amazon S3 でのオブジェクトレベルのロギングを有効にすると、S3 CloudTrail イベントと Lake Formation CloudTrail イベントを結合するクエリを実行して、S3 バケットにアクセスしたアカウントを特定することができます。

トピック

- クロスアカウント CloudTrail ログにプリンシパルアイデンティティを含める
- Amazon S3 クロスアカウントアクセスの CloudTrail ログのクエリ

クロスアカウント CloudTrail ログにプリンシパルアイデンティティを含める

デフォルトでは、共有リソース受信者のログに追加され、リソース所有者のログにコピーされたクロスアカウント CloudTrail イベントには、外部アカウントプリン AWS シパルのプリンシパル ID のみが含まれ、プリンシパル (プリンシパル ARN) の人間が読み取り可能な Amazon リソースネーム (ARN) は含まれません。同じ組織またはチーム内などの信頼できる境界範囲内でリソースを共有するときは、CloudTrail イベントにプリンシパル ARN を含めることをオプトインできます。そうする

ことで、リソース所有者アカウントは、アカウントが所有するリソースにアクセスする受領者アカウントのプリンシパルを追跡できるようになります。

> **Important**
>
> 共有リソースの受領者として独自の CloudTrail ログ内のイベントのプリンシパル ARN を表示するには、所有者アカウントとプリンシパル ARN を共有することをオプトインする必要があります。
> リソースリンク経由でデータアクセスが行われる場合、リソースリンクへのアクセスと、ターゲットリソースへのアクセスの 2 つのイベントが、共有リソース受領者のアカウントにログに記録されます。リソースリンクアクセスのイベントには、プリンシパル ARN が含まれています。オプトインされなかった場合、ターゲットリソースアクセスのイベントにプリンシパル ARN は含まれません。リソースリンクアクセスイベントは、所有者アカウントにコピーされません。

以下は、デフォルトのクロスアカウント CloudTrail イベント (オプトインなし) からの抜粋です。データアクセスを実行するアカウントは 1111-2222-3333 です。これは、呼び出し側のアカウントとリソース所有者アカウントの両方に表示されるログです。クロスアカウントの場合、Lake Formation は両方のアカウントにログを入力します。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId": "AROAQGFTBBBGOBWV2EMZA:GlueJobRunnerSession",
        "accountId": "111122223333"
    },
    "eventSource": "lakeformation.amazonaws.com",
    "eventName": "GetDataAccess",
...
...
    "additionalEventData": {
        "requesterService": "GLUE_JOB",
        "lakeFormationRoleSessionName": "AWSLF-00-GL-111122223333-G13T0Rmng2"
    },
...
}
```

共有リソースのコンシューマーとしてプリンシパル ARN を含めることをオプトインすると、この抜粋は以下のようになります。lakeFormationPrincipal フィールドは、Amazon Athena、Amazon Redshift Spectrum、または AWS Glue ジョブを使用してクエリを実行するエンドロールまたはユーザーを表します。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId": "AROAQGFTBBBGOBWV2EMZA:GlueJobRunnerSession",
        "accountId": "111122223333"
    },
    "eventSource": "lakeformation.amazonaws.com",
    "eventName": "GetDataAccess",
...
...
    "additionalEventData": {
        "requesterService": "GLUE_JOB",
        "lakeFormationPrincipal": "arn:aws:iam::111122223333:role/ETL-Glue-Role",
        "lakeFormationRoleSessionName": "AWSLF-00-GL-111122223333-G13T0Rmng2"
    },
...
}
```

クロスアカウント CloudTrail ログにプリンシパル ARN を含めることをオプトインする

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   Administrator ユーザー、または Administrator Access の IAM ポリシーを持つユーザーとしてサインインします。
2. ナビゲーションペインで [Settings] (設定) を選択します。
3. データカタログ設定ページの「 のデフォルトアクセス許可 AWS CloudTrail」セクションで、リソース所有者に 1 つ以上の AWS リソース所有者アカウント ID を入力します。 IDs

   各アカウント ID の後で Enter キーを押します。
4. [Save] (保存) を選択します。

   これで、共有リソース受領者とリソース所有者両方のログに保存されるクロスアカウント CloudTrail イベントに、プリンシパル ARN が含まれるようになりました。

Amazon S3 クロスアカウントアクセスの CloudTrail ログのクエリ

共有リソース所有者は、S3 CloudTrail ログをクエリして、Amazon S3 バケットにアクセスしたアカウントを特定することができます (Amazon S3 でオブジェクトレベルのロギングが有効化されている場合)。これは、Lake Formation に登録した S3 ロケーションのみに適用されます。共有リソースのコンシューマーが Lake Formation CloudTrail ログにプリンシパル ARN を含めることをオプトインする場合は、バケットにアクセスしたロールまたはユーザーを特定することができます。

を使用してクエリを実行する場合 Amazon Athena、セッション名プロパティで Lake Formation CloudTrail イベントと S3 CloudTrail イベントを結合できます。クエリは、Lake Formation イベントを `eventName="GetDataAccess"` で、S3 イベントを `eventName="Get Object"` または `eventName="Put Object"` でフィルタリングすることもできます。

以下は、登録された S3 ロケーションのデータに対するアクセスが行われた Lake Formation クロスアカウント CloudTrail イベントからの抜粋です。

```
{
  "eventSource": "lakeformation.amazonaws.com",
  "eventName": "GetDataAccess",
  ..............
  ..............
  "additionalEventData": {
    "requesterService": "GLUE_JOB",
    "lakeFormationPrincipal": "arn:aws:iam::111122223333:role/ETL-Glue-Role",
    "lakeFormationRoleSessionName": "AWSLF-00-GL-111122223333-B8JSAjo5QA"
   }
}
```

`lakeFormationRoleSessionName` キーの値である `AWSLF-00-GL-111122223333-B8JSAjo5QA` は、S3 CloudTrail イベントの `principalId` キーにあるセッション名と結合させることができます。以下は、S3 CloudTrail イベントからの抜粋です。これには、セッション名の口ケーションが表示されています。

```
{
   "eventSource": "s3.amazonaws.com",
   "eventName": "Get Object"
   ..............
   ..............
   "principalId": "AROAQSOX5XXUR7D6RMYLR:AWSLF-00-GL-111122223333-B8JSAjo5QA",
```

```
   "arn": "arn:aws:sets::111122223333:assumed-role/Deformationally/AWSLF-00-
GL-111122223333-B8JSAjo5QA",
   "session Context": {
     "session Issuer": {
       "type": "Role",
       "principalId": "AROAQSOX5XXUR7D6RMYLR",
       "arn": "arn:aws:iam::111122223333:role/aws-service-role/
lakeformation.amazonaws.com/Deformationally",
       "accountId": "111122223333",
       "user Name": "Deformationally"
     },
   ..............
   ..............
}
```

セッション名は以下のような形式になります。

```
AWSLF-<version-number>-<query-engine-code>-<account-id->-<suffix>
```

version-number

この形式のバージョンは、現在 00 です。セッション名の形式が変更される場合、次のバージョンは 01 になります。

query-engine-code

データにアクセスしたエンティティを示します。現在の値は次のとおりです。

GL	AWS Glue ETLジョブ
AT	Athena
RE	Amazon Redshift Spectrum

account-id

Lake Formation から認証情報をリクエストした AWS アカウント ID。

suffix

ランダムに生成された文字列。

AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理

AWS Glue または AWS Lake Formationを使用することで、Data Catalog リソースと基盤となるデータに対するクロスアカウントアクセス権を付与することが可能です。

AWS Glue では、データカタログリソースポリシーを作成または更新することでクロスアカウント許可を付与します。Lake Formation では、Lake Formation の `GRANT/REVOKE` 許可モデルと、`Grant Permissions` API 操作を使用することによって、クロスアカウント許可を付与します。

> **Tip**
>
> データレイクをセキュア化するには、Lake Formation 許可のみに頼ることをお勧めします。

Lake Formation のクロスアカウント付与を表示するには、Lake Formation コンソールまたは AWS Resource Access Manager (AWS RAM) コンソールを使用します。ただし、これらのコンソールページには、AWS Glue Data Catalog リソースポリシーによって付与されたクロスアカウント許可が表示されません。同様に、AWS Glue コンソールの [Settings] (設定) ページを使用して Data Catalog リソースポリシー内のクロスアカウント許可を表示することはできますが、そのページに Lake Formation を使用して付与されたクロスアカウント許可は表示されません。

クロスアカウント許可を表示および管理するときに付与を見落とさないようにするため、Lake Formation と AWS Glue では、以下のアクションを実行して、Lake Formation と AWS Glue の両方によるクロスアカウント付与を認識しており、それらを許可していることを示す必要があります。

AWS Glue Data Catalog リソースポリシーを使用してクロスアカウント許可を付与する場合

アカウント (付与者アカウントまたはプロデューサーアカウント) が、 AWS RAM を使用してリソースを共有するクロスアカウント付与を行っていない場合は、データカタログリソースポリシーを通常どおり AWS Glue に保存できます。ただし、 AWS RAM リソース共有を含む許可がすでに作成されている場合は、リソースポリシーの保存が成功するように、次のいずれかを実行する必要があります。

- AWS Glue コンソールの [Settings] (設定) ページでリソースポリシーを保存するときは、ポリシー内の許可が Lake Formation コンソールを使用して付与された許可に追加されることを示す警告が、コンソールに表示されます。[Proceed] (続行) を選択してポリシーを保存する必要があります。

- glue:PutResourcePolicy API オペレーションを使用してリソースポリシーを保存するときは、EnableHybrid フィールドを「TRUE」(型 = 文字列) に設定する必要があります。

  既存のリソースポリシーを更新するには、glue:GetResourcePolicy API オペレーションを使用してまず現在のポリシーを取得し、必要に応じて変更してから glue:PutResourcePolicy を呼び出します。

  > **Note**
  >
  > クロスアカウントアクセス用の AWS Glue リソースポリシーを作成するときは、特定のユースケースに必要な最小限のアクセス許可のみを付与します。

  詳細については、「AWS Glue デベロッパーガイド」の「PutResourcePolicy アクション (Python: put_resource_policy)」を参照してください。

Lake Formation の名前付きリソース方式を使用してクロスアカウント許可を付与する場合

アカウント (プロデューサーアカウント) にデータカタログリソースポリシーがない場合、Lake Formation クロスアカウント付与は通常どおりに実行されます。ただし、Data Catalog リソースポリシーが存在する場合は、以下のステートメントをポリシーに追加して、クロスアカウント付与が名前付きリソース方式で行われた場合でもそれらが成功することを許可する必要があります。*<region>* を有効なリージョン名に置き換え、*<account-id>* を AWS アカウント ID (プロデューサーアカウント ID) に置き換えます。

```
{
  "Effect": "Allow",
  "Action": [
    "glue:ShareResource"
  ],
  "Principal": {"Service": [
    "ram.amazonaws.com"
  ]},
  "Resource": [
    "arn:aws:glue:<region>:<account-id>:table/*/*",
    "arn:aws:glue:<region>:<account-id>:database/*",
    "arn:aws:glue:<region>:<account-id>:catalog"
  ]
}
```

この追加のステートメントがないと、Lake Formation 許可は成功しますが、ブロックされ AWS RAM、受信者アカウントは付与されたリソースにアクセスできません。

> **Important**
>
> クロスアカウント付与を実行するために Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方式も使用している場合、少なくとも「前提条件」で指定されている許可がある Data Catalog リソースポリシーが必要です。

以下も参照してください。

- 「メタデータのアクセスコントロール」(名前付きリソース方式と Lake Formation のタグベースのアクセスコントロール (LF-TBAC) 方式の説明)
- 共有 Data Catalog テーブルとデータベースの表示
- 「AWS Glue デベロッパーガイド」の「AWS Glue コンソールでのデータカタログ設定の使用」
- 「AWS Glue デベロッパーガイド」の「クロスアカウントアクセス許可の付与」(データカタログリソースポリシーのサンプル)

GetResourceShares API 操作を使用したすべてのクロスアカウント付与の表示

企業がリソース AWS Glue Data Catalog ポリシーと Lake Formation 許可の両方を使用してクロスアカウント許可を付与する場合、すべてのクロスアカウント許可を 1 か所に表示する唯一の方法は、`glue:GetResourceShares` API オペレーションを使用することです。

名前付きリソースメソッドを使用してアカウント間で Lake Formation アクセス許可を付与すると、AWS Resource Access Manager (AWS RAM) は AWS Identity and Access Management (IAM) リソースポリシーを作成し、 AWS アカウントに保存します。このポリシーは、リソースへのアクセスに必要なアクセス許可を付与します。 は、クロスアカウント付与ごとに個別のリソースポリシー AWS RAM を作成します。`glue:GetResourceShares` API 操作を使用することで、これらすべてのポリシーを表示することができます。

> **Note**
>
> この操作は、Data Catalog リソースポリシーも返します。ただし、Data Catalog 設定でメタデータ暗号化を有効にし、 AWS KMS キーに対するアクセス許可がない場合、オペレーションは Data Catalog リソースポリシーを返しません。

すべてのクロスアカウント付与を表示する

- 次のコマンドを入力します AWS CLI 。

```
aws glue get-resource-policies
```

データベースのテーブルに対するアクセス許可を AWS アカウント 1111-2222-3333 に付与するときに が AWS RAM 作成および保存するリソースポリシーの例tdb1を次に示します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "glue:GetTable",
         "glue:GetTables",
         "glue:GetTableVersion",
         "glue:GetTableVersions",
         "glue:GetPartition",
         "glue:GetPartitions",
         "glue:BatchGetPartition",
         "glue:SearchTables"
       ],
      "Principal": {"AWS": [
        "111122223333"
      ]},
      "Resource": [
      "arn:aws:glue:us-east-1:111122223333:table/db1/t"
     ]
    }
```

```
  ]
}
```

以下も参照してください。

- AWS Glue デベロッパーガイドの「GetResourceShares アクション (Python: get_resource_policies)」

共有 Data Catalog テーブルとデータベースへのアクセスと表示

データレイク管理者とアクセス許可が付与されたプリンシパルの場合、 AWS アカウントと共有されているリソースは、アカウント内のリソースであるかのように Data Catalog に表示されます。コンソールには、リソースを所有するアカウントが表示されます。

アカウントと共有されているリソースは、Lake Formation コンソールを使用することで表示できます。 AWS Resource Access Manager (AWS RAM) コンソールを使用して、 アカウントと共有されているリソースと、名前付きリソースメソッドを使用して他の AWS アカウントと共有したリソースの両方を表示することもできます。

Important

名前付きリソースメソッドを使用して Data Catalog リソースに対するクロスアカウントアクセス許可をアカウントまたは AWS 組織に付与すると、Lake Formation は AWS Resource Access Manager (AWS RAM) サービスを使用してリソースを共有します。アカウントが付与アカウントと同じ AWS 組織にある場合、共有リソースはすぐに使用できます。
ただし、アカウントが同じ組織にない場合、 はリソース共有を承認または拒否するための招待をアカウント AWS RAM に送信します。次に、共有リソースを使用できるようにするには、アカウントのデータレイク管理者が AWS RAM コンソールまたは CLI を使用して招待を受け入れる必要があります。
Lake Formation コンソールには、承諾を待っている AWS RAM リソース共有の招待がある場合にアラートが表示されます。 AWS RAM 招待の表示を許可されたユーザーのみがアラートを受け取ります。

以下も参照してください。

- AWS アカウント間でのデータカタログテーブルとデータベースの共有
- Lake Formation でのクロスアカウントデータ共有
- 共有テーブルの基盤となるデータへのアクセス
- 「メタデータのアクセスコントロール」(リソースを共有するための名前付きリソース方式と LF-TBAC 方式に関する情報)

トピック

- からのリソース共有の招待の承諾 AWS RAM
- 共有 Data Catalog テーブルとデータベースの表示

からのリソース共有の招待の承諾 AWS RAM

Data Catalog リソースが AWS アカウントと共有されており、アカウントが共有アカウントと同じ AWS 組織内にない場合、 AWS Resource Access Manager () からのリソース共有の招待を受け入れるまで、共有リソースにアクセスすることはできませんAWS RAM。データレイク管理者として、まず保留中の招待 AWS RAM をクエリしてから、招待を受け入れる必要があります。

AWS RAM コンソール、API、または AWS Command Line Interface (AWS CLI) を使用して、招待を表示および承諾できます。

からリソース共有の招待を表示して受け入れるには AWS RAM (コンソール)

1. リソース共有の招待を表示して承諾するために必要な AWS Identity and Access Management (IAM) アクセス許可があることを確認します。

   データレイク管理者に推奨される IAM ポリシーについては、「the section called “データレイク管理者の許可”」を参照してください。
2. AWS RAM ユーザーガイドの「招待の受け入れと拒否」にある手順を実行します。

(AWS RAM AWS CLI) からリソース共有の招待を表示して受け入れるには

1. リソース共有の招待を表示して承諾するために必要な AWS Identity and Access Management (IAM) アクセス許可があることを確認します。

データレイク管理者に推奨される IAM ポリシーについては、「the section called “データレイク管理者の許可”」を参照してください。

2. 以下のコマンドを入力して、保留中のリソース共有招待を表示します。

```
aws ram get-resource-share-invitations
```

出力は以下のようになります。

```
{
    "resourceShareInvitations": [
        {
            "resourceShareInvitationArn": "arn:aws:ram:us-
east-1:111122223333:resource-share-invitation/a93aa60a-1bd9-46e8-96db-
a4e72eec1d9f",
            "resourceShareName": "111122223333-123456789012-uswuU",
            "resourceShareArn": "arn:aws:ram:us-east-1:111122223333:resource-
share/2a4ab5fb-d859-4751-84f7-8760b35fc1fe",
            "senderAccountId": "111122223333",
            "receiverAccountId": "123456789012",
            "invitationTimestamp": 1589576601.79,
            "status": "PENDING"
        }
    ]
}
```

PENDING のステータスに注意してください。

3. resourceShareInvitationArn キーの値をクリップボードにコピーします。

4. その値を以下のコマンドに貼り付けて *<invitation-arn>* を置き換え、コマンドを入力します。

```
aws ram accept-resource-share-invitation --resource-share-invitation-
arn <invitation-arn>
```

出力は以下のようになります。

```
{
    "resourceShareInvitations": [
        {
```

```
            "resourceShareInvitationArn": "arn:aws:ram:us-
east-1:111122223333:resource-share-invitation/a93aa60a-1bd9-46e8-96db-
a4e72eec1d9f",
            "resourceShareName": "111122223333-123456789012-uswuU",
            "resourceShareArn": "arn:aws:ram:us-east-1:111122223333:resource-
share/2a4ab5fb-d859-4751-84f7-8760b35fc1fe",
            "senderAccountId": "111122223333",
            "receiverAccountId": "123456789012",
            "invitationTimestamp": 1589576601.79,
            "status": "ACCEPTED"
        }
    ]
}
```

ACCEPTED のステータスに注意してください。

共有 Data Catalog テーブルとデータベースの表示

アカウントと共有されているリソースは、Lake Formation コンソール、または AWS CLI を使用することで表示できます。 AWS Resource Access Manager (AWS RAM) コンソールまたは CLI を使用して、アカウントと共有されているリソースと、他の AWS アカウントと共有しているリソースの両方を表示することもできます。

Lake Formation コンソールを使用して共有リソースを表示する

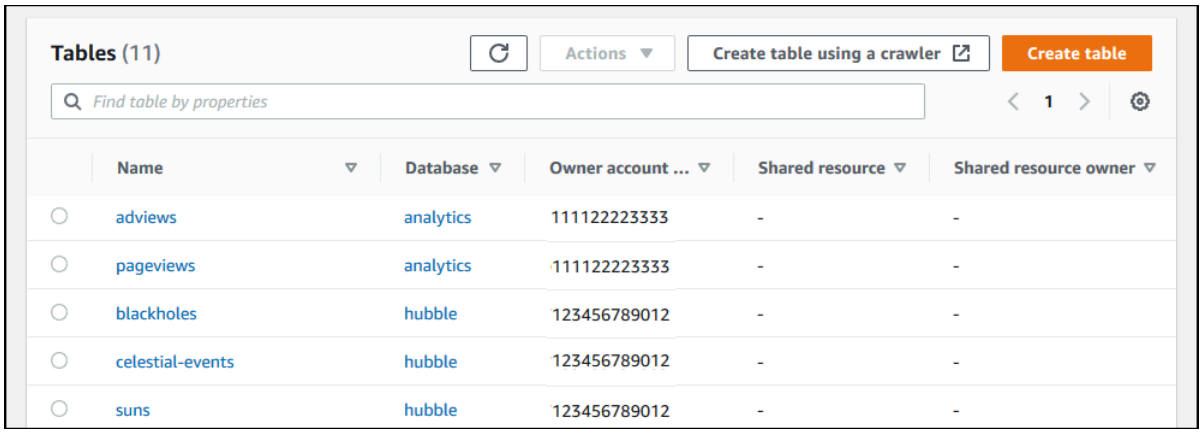
1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) を開きます。

   データレイク管理者、または共有テーブルに対する許可を付与されたユーザーとしてサインインします。
2. AWS アカウントと共有されているリソースを表示するには、次のいずれかを実行します。

   - アカウントと共有されたテーブルを表示するには、ナビゲーションペインで [Tables] (テーブル) を選択します。
   - アカウントと共有されたデータベースを表示するには、ナビゲーションペインで [Databases] (データベース) を選択します。

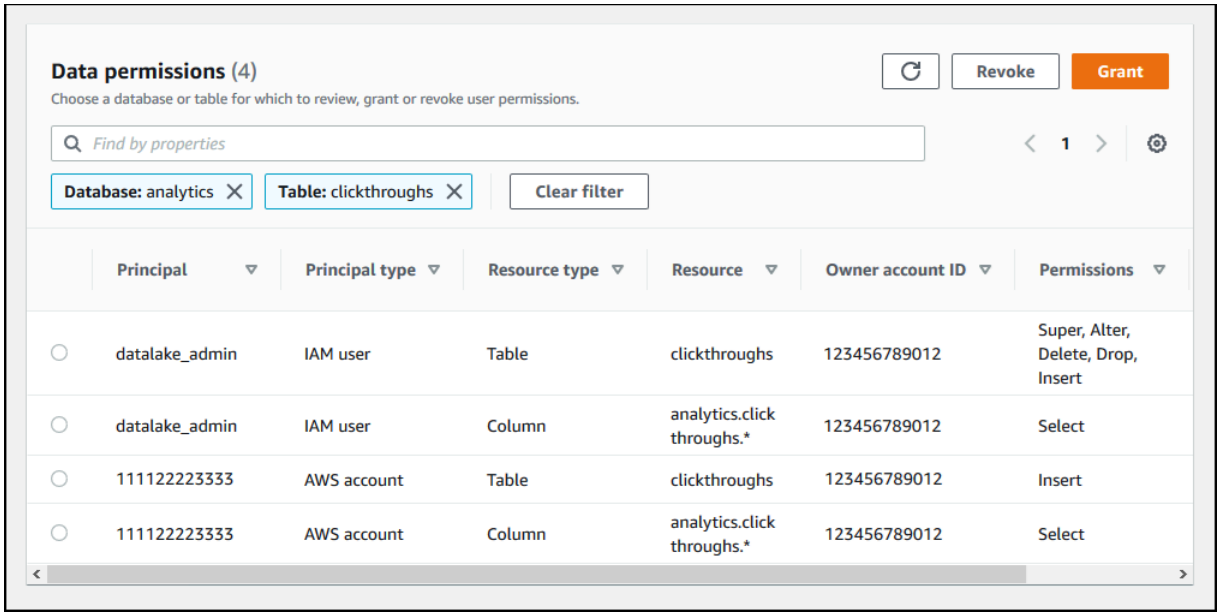
   コンソールに、アカウント内のデータベースまたはテーブル、およびアカウントと共有されたデータベースまたはテーブルの両方のリストが表示されます。アカウントと共有されたリソース

については、コンソールの [Owner account ID] (所有者アカウント ID) 列に所有者の AWS アカウント ID が表示されます (以下のスクリーンショットでは 3 番目の列)。



3. 他の AWS アカウントまたは組織と共有したリソースを表示するには、ナビゲーションペインでデータアクセス許可を選択します。

   共有したリソースは、以下の画像にあるように [Data permissions] (データの許可) ページにリストされ、[Principal] (プリンシパル) 列に外部アカウント番号が表示されます。



AWS RAM コンソールを使用して共有リソースを表示するには

1. を使用して共有リソースを表示するために必要な AWS Identity and Access Management (IAM) アクセス許可があることを確認します AWS RAM。

少なくとも `ram:ListResources` 許可が必要です。この許可は、AWS マネージドポリシーの `AWSLakeFormationCrossAccountManager` に含まれています。

2. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/ram/home で AWS RAM コンソールを開きます。
3. 次のいずれかを行います。
   - ユーザーが共有したリソースを表示するには、ナビゲーションペインの [Shared by me] (自分が共有) で [Shared resources] (共有リソース) を選択します。
   - ユーザーと共有されているリソースを表示するには、ナビゲーションペインの [Shared by me] (自分と共有) で [Shared resources] (共有リソース) を選択します。

リソースリンクの作成

リソースリンクは、メタデータデータベースとテーブルへのリンクである Data Catalog オブジェクトです。通常は、他の AWS アカウントの共有データベースとテーブルへのリンクです。これにより、すべての AWS リージョンでデータレイク内のデータへのクロスアカウントアクセスが可能になります。

> **Note**
>
> Lake Formation は、 AWS リージョン間でのデータカタログテーブルのクエリをサポートしています。異なる AWS リージョンの共有データベースとテーブルを指すリソースリンクをそれらのリージョンに作成することで、任意のリージョンから Data Catalog データベースとテーブルにアクセスできます。

トピック

- Lake Formation でのリソースリンクの仕組み
- 共有 Data Catalog テーブルへのリソースリンクの作成
- 共有 Data Catalog データベースへのリソースリンクの作成
- AWS Glue API でのリソースリンク処理

Lake Formation でのリソースリンクの仕組み

リソースリンクは、ローカルまたは共有のデータベースやテーブルへのリンクである データカタログオブジェクトです。データベースまたはテーブルへのリソースリンクを作成すると、そのデータベース名やテーブル名を使用する場所ならどこでもリソースリンク名を使用することができます。テーブルのリソースリンクは、`glue:GetTables()` によってユーザーが所有するテーブル、またはユーザーと共有されたテーブルとともに返され、Lake Formation コンソールの [Tables] (テーブル) ページにエントリとして表示されます。データベースへのリソースリンクも同様に機能します。

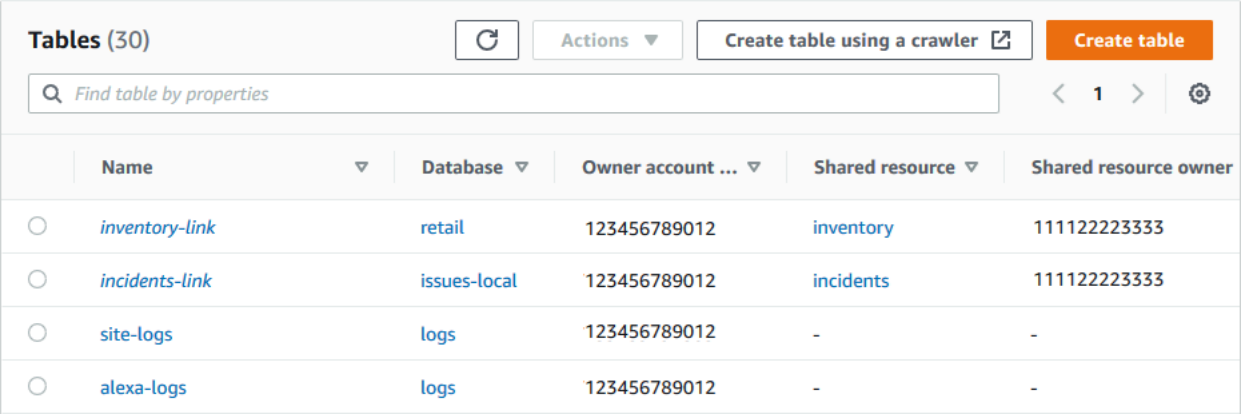
データベースまたはテーブルへのリソースリンクを作成すると、以下を実行できるようになります。

- Data Catalog 内のデータベースまたはテーブルに異なる名前を割り当てる。これは、異なる AWS アカウントが同じ名前のデータベースまたはテーブルを共有する場合、またはアカウント内の複数のデータベースに同じ名前のテーブルがある場合に特に便利です。
- 任意の AWS リージョンから Data Catalog データベースとテーブルにアクセスするには、別のリージョンのデータベースとテーブルを指すリソースリンクをそれらのリージョンに作成します。ソースデータやメタデータを Glue データカタログにコピーしなくても、これらのリソースリンクを Athena や Amazon EMR で使用してどのリージョンでもクエリを実行し、 AWS Glue ETL Spark ジョブを実行できます。
- Amazon Athena や Amazon Redshift Spectrum などの統合 AWS サービスを使用して、共有データベースやテーブルにアクセスするクエリを実行します。統合サービスには、アカウントをまたいでデータベースやテーブルに直接アクセスできないものがありますが、アカウントにある他のアカウントのデータベースやテーブルへのリソースリンクにアクセスすることは可能です。

> **Note**
>
> AWS Glue 抽出、変換、ロード (ETL) スクリプトで共有データベースやテーブルを参照するためにリソースリンクを作成する必要はありませんが、複数の AWS アカウントが同じ名前のデータベースやテーブルを共有している場合の曖昧さを回避するために、リソースリンクを作成して使用するか、ETL 操作の呼び出し時にカタログ ID を指定することができます。

以下は、2 つのリソースリンクが表示されている Lake Formation コンソールの [Tables] (テーブル) ページの例です。リソースリンクの名前は、常にイタリック体で表示されます。各リソースリンクは、リンクされた共有リソースの名前と所有者と共に表示されます。この例では、 AWS アカウント 1111-2222-3333 のデータレイク管理者が、 テーブル`inventory`と `incidents`テーブルをアカウン

ト 1234-5678-9012 と共有しました。共有後、そのアカウントのユーザーがそれらの共有テーブルへのリソースリンクを作成しました。



以下は、リソースリンクに関する注意点と制限です。

- リソースリンクでは、共有テーブルの基盤となるデータをクエリするために、Athena および Redshift Spectrum などの統合サービスを有効にすることが必要になります。これらの統合サービスでのクエリは、リソースリンク名に対して作成されます。
- テーブルが含まれるデータベースの [Use only IAM access control for new tables in this database] (このデータベースの新しいテーブルには IAM アクセス制御のみを使用する) 設定がオフになっていることを前提とすると、データベースを表示してそれにアクセスできるのは、リソースリンクを作成したプリンシパルのみになります。アカウント内の他のプリンシパルがリソースリンクにアクセスできるようにするには、それに対する DESCRIBE 許可を付与します。他のユーザーがリソースリンクをドロップできるようにするには、それに対する DROP 許可を付与します。データレイク管理者は、アカウント内のすべてのリソースリンクにアクセスできます。別のプリンシパルが作成したリソースリンクをドロップするには、まずデータレイク管理者がリソースリンクに対する DROP 許可を管理者自身に付与する必要があります。詳細については、「Lake Formation 許可のリファレンス」を参照してください。

  > **Important**
  >
  > リソースリンクに対する許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルに対する許可は付与されません。ターゲットに対する許可は、別途付与する必要があります。

- リソースリンクを作成するには、Lake Formation CREATE_TABLEまたは アクセスCREATE_DATABASE許可と、 glue:CreateTableまたは glue:CreateDatabase AWS Identity and Access Management (IAM) アクセス許可が必要です。

- ローカル (所有) Data Catalog リソースと、 AWS アカウントと共有されているリソースへのリソースリンクを作成できます。
- リソースリンクを作成するときに、ターゲット共有リソースが存在するかどうか、またはそのリソースに対するクロスアカウント許可があるかどうかを確認するためのチェックは実行されません。これは、リソースリンクと共有リソースを任意の順序で作成できるようにします。
- リソースリンクを削除しても、リンクされた共有リソースはドロップされません。共有リソースをドロップしても、そのリソースへのリソースリンクは削除されません。
- リソースリンクチェーンを作成することが可能ですが、API は最初のリソースリンクのみを使用するので、作成する価値はありません。

以下も参照してください。

- データカタログリソースに対するアクセス許可の付与

共有 Data Catalog テーブルへのリソースリンクの作成

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、任意の AWS リージョンの共有テーブルへのリソースリンクを作成できますAWS CLI。

共有テーブルへのリソースリンクを作成するには (コンソール)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。リソースリンクの保存先になるデータベースに対する Lake Formation CREATE_TABLE 許可を持つプリンシパルとしてサインインします。
2. ナビゲーションペインの [データカタログ] で、[テーブル] を選択し、[作成]、[リソースリンク] の順に選択します。
3. [リソースリンクの作成] ページで、以下の情報を入力します。

   [Resource link name] (リソースリンク名)

   テーブル名と同じルールに従う名前を入力します。名前は、ターゲット共有テーブルと同じものにすることができます。

   [Database] (データベース)

   リソースリンクの保存先になるローカル Data Catalog 内のデータベースです。

[共有テーブル所有者のリージョン]

別のリージョンでリソースリンクを作成する場合は、ターゲット共有テーブルのリージョンを選択します。

[Shared table] (共有テーブル)

リストから共有テーブルを選択するか、ローカル (所有する) または共有テーブル名を入力します。

このリストには、アカウントと共有されているすべてのテーブルが含まれています。各テーブルにリストされているデータベースと所有者アカウント ID に注意してください。アカウントと共有されていることが分かっているテーブルが表示されない場合は、以下を確認してください。

- データレイク管理者ではない場合は、データレイク管理者からそのテーブルに対する Lake Formation 許可が付与されていることを確認します。
- データレイク管理者であり、アカウントが付与元のアカウントと同じ AWS 組織にない場合は、テーブルに関する AWS Resource Access Manager (AWS RAM) リソース共有招待を承諾していることを確認します。詳細については、「からのリソース共有の招待の承諾 AWS RAM」を参照してください。

[Shared table's database] (共有テーブルのデータベース)

リストから共有テーブルを選択した場合、このフィールドには外部アカウントにある共有テーブルのデータベースが入力されます。入力されていないときは、ローカルデータベース (ローカルテーブルへのリソースリンクの場合)、または外部アカウントにある共有テーブルのデータベースを入力します。

[Shared table owner] (共有テーブル所有者)

リストから共有テーブルを選択した場合、このフィールドには共有テーブルの所有者アカウント ID が入力されます。それ以外の場合は、 AWS アカウント ID (ローカルテーブルへのリソースリンク用) またはテーブルを共有した AWS アカウントの ID を入力します。

4. [Create] (作成) を選択して、リソースリンクを作成します。

その後、[Tables] (テーブル) ページの [Name] (名前) 列でリソースリンク名を確認することができます。

5. (オプション) リンクを表示してリンク先のテーブルにアクセスできることが必要なプリンシパルに対して、リソースリンクへの Lake Formation の `DESCRIBE` 許可を付与します。

ただし、リソースリンクへのアクセス許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルへのアクセス許可は付与されません。Athena でテーブル/リソースリンクが表示されるようにするには、ターゲットのデータベースへのアクセス許可を別途付与する必要があります。

同じリージョン内の共有テーブルへのリソースリンクを作成するには (AWS CLI)

1. 以下のようなコマンドを入力します。

```
aws glue create-table --database-name myissues --table-input
 '{"Name":"my_customers","TargetTable":
{"CatalogId":"111122223333","DatabaseName":"issues","Name":"customers"}}'
```

このコマンドは、 AWS アカウント 1111-2222-3333 のデータベース issues にある共有テーブル customers に my_customers という名前のリソースリンクを作成します。リソースリンクは、ローカルデータベース myissues に保存されます。

2. (オプション) リンクを表示してリンク先のテーブルにアクセスできることが必要なプリンシパルに対して、リソースリンクへの Lake Formation の DESCRIBE 許可を付与します。

リソースリンクへのアクセス許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルへのアクセス許可は付与されません。Athena でテーブル/リソースリンクが表示されるようにするには、ターゲットのデータベースへのアクセス許可を別途付与する必要があります。

異なるリージョン内の共有テーブルへのリソースリンクを作成するには (AWS CLI)

1. 以下のようなコマンドを入力します。

```
aws glue create-table --region eu-west-1 --cli-input-json '{
    "CatalogId": "111122223333",
    "DatabaseName": "ireland_db",
    "TableInput": {
        "Name": "rl_useast1salestb_ireland",
        "TargetTable": {
            "CatalogId": "444455556666",
            "DatabaseName": "useast1_salesdb",
            "Region": "us-east-1",
            "Name":"useast1_salestb"
```

```
        }
    }
}'
```

このコマンドは、米国東部 (バージニア北部) リージョンrl_useast1salestb_irelandの AWS アカウント 444455556666 useast1_salesdbのデータベースにある共有テーブルにuseast1_salestb、 という名前のリソースリンクを作成します。リソースリンクは、ローカルデータベース ireland_db に保存されます。

2. リンクを表示してリンク先にアクセスできることが必要なプリンシパルに対して、Lake Formation の DESCRIBE 許可を付与します。

   リソースリンクへのアクセス許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルへのアクセス許可は付与されません。Athena でテーブル/リソースリンクが表示されるようにするには、ターゲットのテーブルへのアクセス許可を別途付与する必要があります。

以下も参照してください。

- Lake Formation でのリソースリンクの仕組み
- DESCRIBE

共有 Data Catalog データベースへのリソースリンクの作成

AWS Lake Formation コンソール、API、または AWS Command Line Interface () を使用して、共有データベースへのリソースリンクを作成できますAWS CLI。

共有データベースへのリソースリンクを作成する (コンソール)

1. https://console.aws.amazon.com/lakeformation/ で AWS Lake Formation コンソールを開きます。データレイク管理者またはデータベース作成者としてサインインします。

   データベース作成者は、Lake Formation の CREATE_DATABASE 許可を付与されたプリンシパルです。

2. ナビゲーションペインで、[データベース] を選択し、[作成]、[リソースリンク] の順に選択します。

3. [リソースリンクの作成] ページで、以下の情報を入力します。

[Resource link name] (リソースリンク名)

データベース名と同じルールに従う名前を入力します。名前は、ターゲット共有データベースと同じものにすることができます。

送信先カタログ

データベースリソースリンクの送信先カタログを選択します。

[共有データベース所有者のリージョン]

別のリージョンでリソースリンクを作成する場合は、ターゲットの共有データベースのリージョンを選択します。

[Shared database] (共有データベース)

リストからデータベースを選択するか、ローカル (所有する) または共有データベース名を入力します。

このリストには、アカウントと共有されているすべてのデータベースが含まれています。各データベースにリストされている所有者アカウント ID に注意してください。アカウントと共有されていることが分かっているデータベースが表示されない場合は、以下を確認してください。

- データレイク管理者ではない場合は、データレイク管理者からそのデータベースに対する Lake Formation 許可が付与されていることを確認します。
- データレイク管理者であり、アカウントが付与元のアカウントと同じ AWS 組織にない場合は、データベースに関する AWS Resource Access Manager (AWS RAM) リソース共有招待を承諾していることを確認します。詳細については、「からのリソース共有の招待の承諾 AWS RAM」を参照してください。

[Shared database owner] (共有データベース所有者)

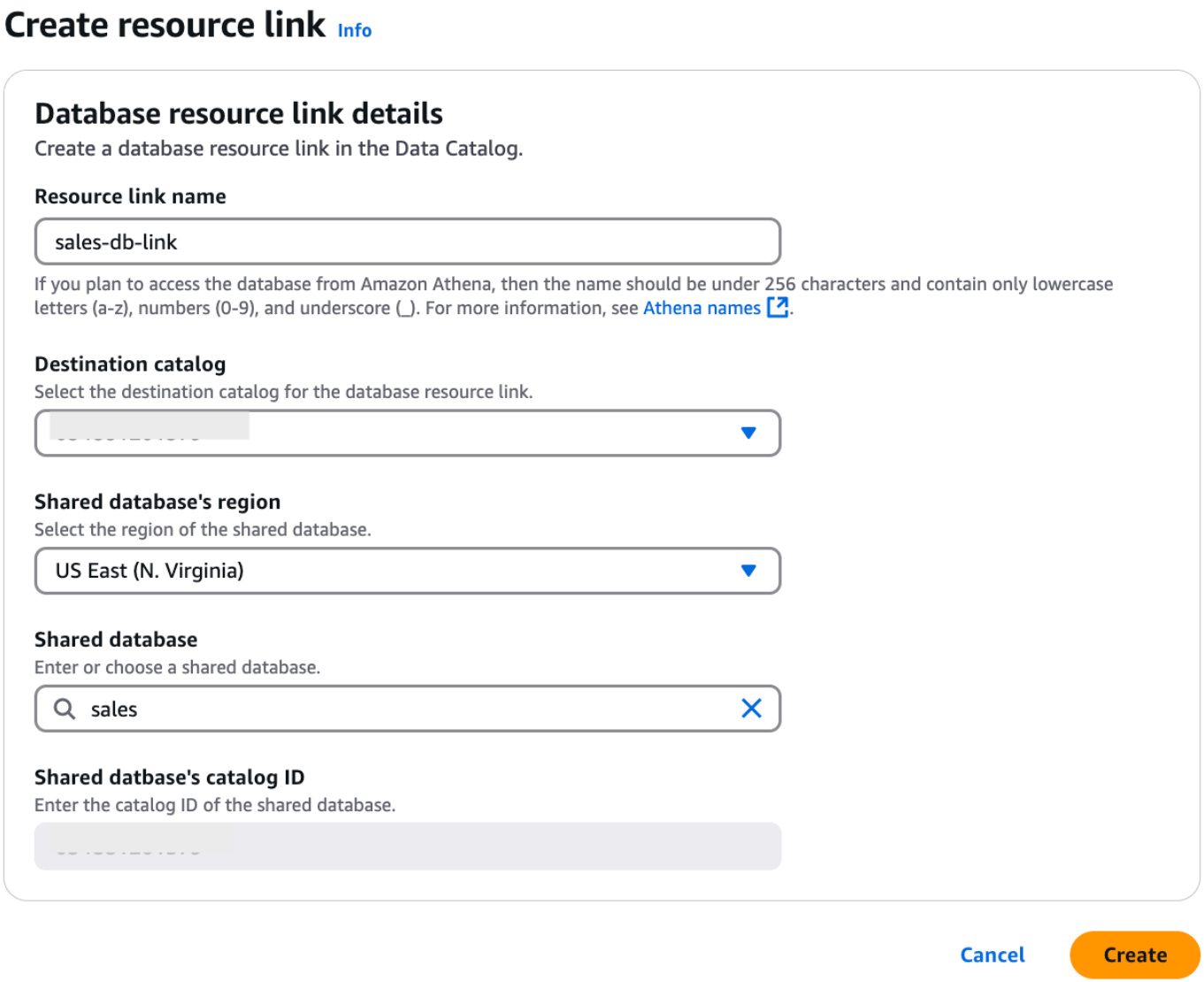
リストから共有データベースを選択した場合、このフィールドには共有データベースの所有者アカウント ID が入力されます。それ以外の場合は、 AWS アカウント ID (ローカルデータベースへのリソースリンク用) またはデータベースを共有した AWS アカウントの ID を入力します。

共有データベースのカタログ ID

共有データベースのカタログ ID を入力します。別の AWS アカウントから共有されているデータベースへのリソースリンクを作成するときは、このカタログ ID を指定して、どのア

カウントのデータカタログにソースデータベースが含まれているかを特定する必要があります。

ドロップダウンメニューから共有データベースを選択すると、そのデータベースを所有し、共有しているアカウントのカタログ ID が自動的に入力されます。



4. [Create] (作成) を選択して、リソースリンクを作成します。

   その後、[Database] (データベース) ページの [Name] (名前) 列でリソースリンク名を確認することができます。

5. (オプション) リンクを表示してリンク先のデータベースにアクセスできることが必要な、欧州 (アイルランド) リージョンのプリンシパルに対して、リソースリンクへの Lake Formation の DESCRIBE 許可を付与します。

ただし、リソースリンクへのアクセス許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルへのアクセス許可は付与されません。Athena でテーブル/リソースリンクが表示されるようにするには、ターゲットのデータベースへのアクセス許可を別途付与する必要があります。

同じリージョン内の共有データベースへのリソースリンクを作成するには (AWS CLI)

1. 以下のようなコマンドを入力します。

```
aws glue create-database --database-input '{"Name":"myissues","TargetDatabase":
{"CatalogId":"111122223333","DatabaseName":"issues"}}'
```

このコマンドは、 AWS アカウント 1111-2222-3333 にある共有データベース myissuesに issuesという名前のリソースリンクを作成します。

2. (オプション) リンクを表示してリンク先のデータベースにアクセスする許可が必要なプリンシパルに、リソースリンクに対する Lake Formation DESCRIBE アクセス許可を付与します。

ただし、リソースリンクへのアクセス許可を付与しても、ターゲットの (リンクされた) データベースまたはテーブルへのアクセス許可は付与されません。Athena でテーブル/リソースリンクが表示されるようにするには、ターゲットのデータベースへのアクセス許可を別途付与する必要があります。

異なるリージョン内の共有データベースへのリソースリンクを作成するには (AWS CLI)

1. 以下のようなコマンドを入力します。

```
aws glue create-database --region eu-west-1 --cli-input-json '{
    "CatalogId": "111122223333",
    "DatabaseInput": {
      "Name": "rl_useast1shared_irelanddb",
      "TargetDatabase": {
          "CatalogId": "444455556666",
          "DatabaseName": "useast1shared_db",
          "Region": "us-east-1"
       }
    }
}'
```

このコマンドは、欧州 (アイルランド) リージョンrl_useast1shared_irelanddbの AWS アカウント 111122223333 に という名前のリソースリンクを作成しuseast1shared_db、共有データベース を作成します。共有データベース は、米国東部 (バージニア北部) リージョンの AWS アカウント 444455556666 にあります。

2. リンクを表示してリンク先にアクセスできることが必要な、欧州 (アイルランド) リージョンのプリンシパルに対して、Lake Formation の DESCRIBE 許可を付与します。

以下も参照してください。

- Lake Formation でのリソースリンクの仕組み
- DESCRIBE

AWS Glue API でのリソースリンク処理

以下は、AWS Glue Data Catalog API がデータベースおよびテーブルリソースリンクを処理する方法を説明する表です。すべての Get* API オペレーションで、呼び出し側が許可を持つデータベースとテーブルのみが返されます。また、リソースリンクを介してターゲットデータベースまたはテーブルにアクセスする場合、ターゲットとリソースリンクの両方に対する AWS Identity and Access Management (IAM) と Lake Formation の両方のアクセス許可が必要です。リソースリンクに対する必要な Lake Formation 許可は DESCRIBE です。詳細については、「DESCRIBE」を参照してください。

データベースの API オペレーション

API オペレーション:	リソースリンクの処理
CreateDatabase	データベースがリソースリンクである場合、指定されたターゲットデータベースへのリソースリンクを作成します。
UpdateDatabase	指定されたデータベースがリソースリンクである場合、リンクをたどってターゲットデータベースを更新します。リソースリンクを異なるデータベースにリンクするように変更する必要がある場合は、リソースリンクを削除して、新しいリソースリンクを作成する必要があります。

API オペレーション:	リソースリンクの処理
DeleteDatabase	リソースリンクを削除します。リンクされた (ターゲット) データベースは削除しません。
GetDatabase	呼び出し元がターゲットに対する許可を持っている場合、リンクをたどってターゲットのプロパティを返します。それ以外の場合は、リンクのプロパティを返します。
GetDatabases	リソースリンクを含めたデータベースのリストを返します。結果セット内のリソースリンクごとに操作がリンクをたどり、リンクターゲットのプロパティを取得します。アカウントと共有されているデータベースを表示するには、ResourceShareType = ALL を指定する必要があります。

テーブルの API オペレーション

API オペレーション:	リソースリンクの処理
CreateTable	データベースがリソースリンクである場合、データベースリンクをたどってターゲットデータベース内にテーブルを作成します。テーブルがリソースリンクである場合は、操作が指定されたデータベースでリソースリンクを作成します。データベースリソースリンクを経由したテーブルリソースリンクの作成はサポートされていません。
UpdateTable	テーブルまたは指定されたデータベースがリソースリンクである場合、ターゲットテーブルを更新します。テーブルとデータベースの両方がリソースリンクである場合は、操作が失敗します。
DeleteTable	指定されたデータベースがリソースリンクである場合、リンクをたどってターゲットデータベース内のテーブルまたはテーブルリソースリンクを削除します。テーブルがリソースリンクである場合は、操作が指定されたデータベース内のテーブルリソースリンクを削除します。テーブルリソースリンクを削除しても、ターゲットテーブルは削除されません。
BatchDeleteTable	DeleteTable と同じです。

API オペレーション:	リソースリンクの処理
GetTable	指定されたデータベースがリソースリンクである場合、データベースリンクをたどってターゲットデータベース内からテーブルまたはテーブルリソースリンクを返します。そうでない場合、テーブルがリソースリンクであれば、操作がリンクをたどってターゲットテーブルのプロパティを返します。
GetTables	指定されたデータベースがリソースリンクである場合、データベースリンクをたどってターゲットデータベース内からテーブルとテーブルリソースリンクを返します。ターゲットデータベースが別の AWS アカウントの共有データベースである場合、オペレーションはそのデータベースの共有テーブルのみを返します。これは、ターゲットデータベース内のテーブルリソースリンクをたどりません。そうでない場合、指定されたデータベースがローカル (所有する) データベースであれば、操作がローカルデータベース内のすべてのテーブルを返し、各テーブルリソースリンクをたどってターゲットテーブルのプロパティを返します。
SearchTables	テーブルとテーブルリソースリンクを返します。これは、リンクをたどってターゲットテーブルのプロパティを返しません。アカウントと共有されているテーブルを表示するには、ResourceShareType = ALL を指定する必要があります。
GetTableVersion	GetTable と同じです。
GetTableVersions	GetTable と同じです。
DeleteTableVersion	DeleteTable と同じです。
BatchDeleteTableVersion	DeleteTable と同じです。

パーティションの API オペレーション

API オペレーション:	リソースリンクの処理
CreatePartition	指定されたデータベースがリソースリンクである場合、データベースリンクをたどって、ターゲットデータベース内に指定されたテーブルにパーティションを作成します。テーブルがリソースリンクである場合は、操作がリソースリンクをたどってターゲットテーブルにパーティションを作成します。テーブルリソースリンクとデータベースリソースリンクの両方を通じたパーティションの作成はサポートされていません。
BatchCreatePartition	CreatePartition と同じです。
UpdatePartition	指定されたデータベースがリソースリンクである場合、データベースリンクをたどって、ターゲットデータベース内にある指定されたテーブルのパーティションを更新します。テーブルがリソースリンクである場合は、操作がリソースリンクをたどってターゲットテーブルのパーティションを更新します。テーブルリソースリンクとデータベースリソースリンクの両方を通じたパーティションの更新はサポートされていません。
DeletePartition	指定されたデータベースがリソースリンクである場合、データベースリンクをたどって、ターゲットデータベース内にある指定されたテーブルのパーティションを削除します。テーブルがリソースリンクである場合は、操作がリソースリンクをたどってターゲットテーブルのパーティションを削除します。テーブルリソースリンクとデータベースリソースリンクの両方を通じたパーティションの削除はサポートされていません。
BatchDeletePartition	DeletePartition と同じです。
GetPartition	指定されたデータベースがリソースリンクである場合、データベースリンクをたどって指定されたテーブルからのパーティション情報を返します。そうでない場合、テーブルがリソースリンクであれば、操作がリンクをたどってパーティション情報を返します。テーブ

API オペレーション:	リソースリンクの処理
	ルとデータベースの両方がリソースリンクである場合は、空の結果セットが返されます。
GetPartitions	指定されたデータベースがリソースリンクである場合、データベースリンクをたどって、指定されたテーブル内のすべてのパーティションのパーティション情報を返します。そうでない場合、テーブルがリソースリンクであれば、操作がリンクをたどってパーティション情報を返します。テーブルとデータベースの両方がリソースリンクである場合は、空の結果セットが返されます。
BatchGetPartition	GetPartition と同じです。

ユーザー定義関数の API オペレーション

API オペレーション:	リソースリンク処理
(すべての API オペレーション)	データベースがリソースリンクである場合は、リソースリンクをたどり、ターゲットデータベースで操作を実行します。

以下も参照してください。

- Lake Formation でのリソースリンクの仕組み

クロスリージョンのテーブルアクセス

Lake Formation は AWS 、リージョン間でのデータカタログテーブルのクエリをサポートしています。Amazon Athena、Amazon EMR、および AWS Glue ETL を使用して他のリージョンからリージョンのデータにアクセスするには、ソースデータベースとテーブルを指す他のリージョンにリソースリンクを作成します。クロスリージョンのテーブルアクセスでは、基になるデータやメタデータをデータカタログ内にコピーしなくても、複数のリージョンをまたいでデータにアクセスできます。

例えば、リージョン A でプロデューサーアカウントのデータベースやテーブルをコンシューマーアカウントと共有できます。コンシューマーアカウントのデータレイク管理者は、リージョン A でリソース共有の招待を受け入れ、共有リソースへのリソースリンクをリージョン B に作成できます。

コンシューマーアカウント管理者は、リージョン A でアカウントの IAM プリンシパルに対して、共有リソースへのアクセス許可を付与し、リージョン B のリソースリンクへのアクセス許可を付与できます。このリソースリンクを使用して、コンシューマーアカウントのプリンシパルは、リージョン B から共有データにクエリを実行できます。

リージョン A の Amazon S3 データソースをプロデューサーアカウントでホストし、データの場所をリージョン B の中央アカウントに登録することもできます。中央アカウントでデータカタログのリソースを作成し、Lake Formation のアクセス許可を設定して、自分のアカウントまたはリージョン B の外部アカウントとデータを共有できます。クロスリージョン機能により、ユーザーはリソースリンクを使用してリージョン C から、これらのデータカタログのテーブルにアクセスできます。

この機能を使用すると、複数のリージョンをまたいで Apache Hive メタストアにあるフェデレーションデータベースにクエリを実行したり、クエリを実行するときにローカルリージョンのテーブルを別のリージョンのテーブルと結合したりできます。

Lake Formation は、クロスリージョンのテーブルアクセスで以下の機能をサポートしています。

- LF タグベースのアクセス制御
- きめ細かなアクセス制御のアクセス許可
- 適切なアクセス許可を使用した共有データベースやテーブルへの書き込みオペレーション
- アカウントレベルでのクロスアカウントのデータ共有と IAM プリンシパルレベルでの直接データ共有

管理者以外のユーザーでも、`Create_Database` や `Create_Table` アクセス許可があれば、クロスリージョンのリソースリンクを作成できます。

> **Note**
>
> Lake Formation のアクセス許可を適用しなくても、任意のリージョンでクロスリージョンのリソースリンクを作成し、データにアクセスできます。Lake Formation に登録されていない Amazon S3 のソースデータの場合、アクセスは Amazon S3 の IAM アクセス許可ポリシーと AWS Glue アクションによって決まります。

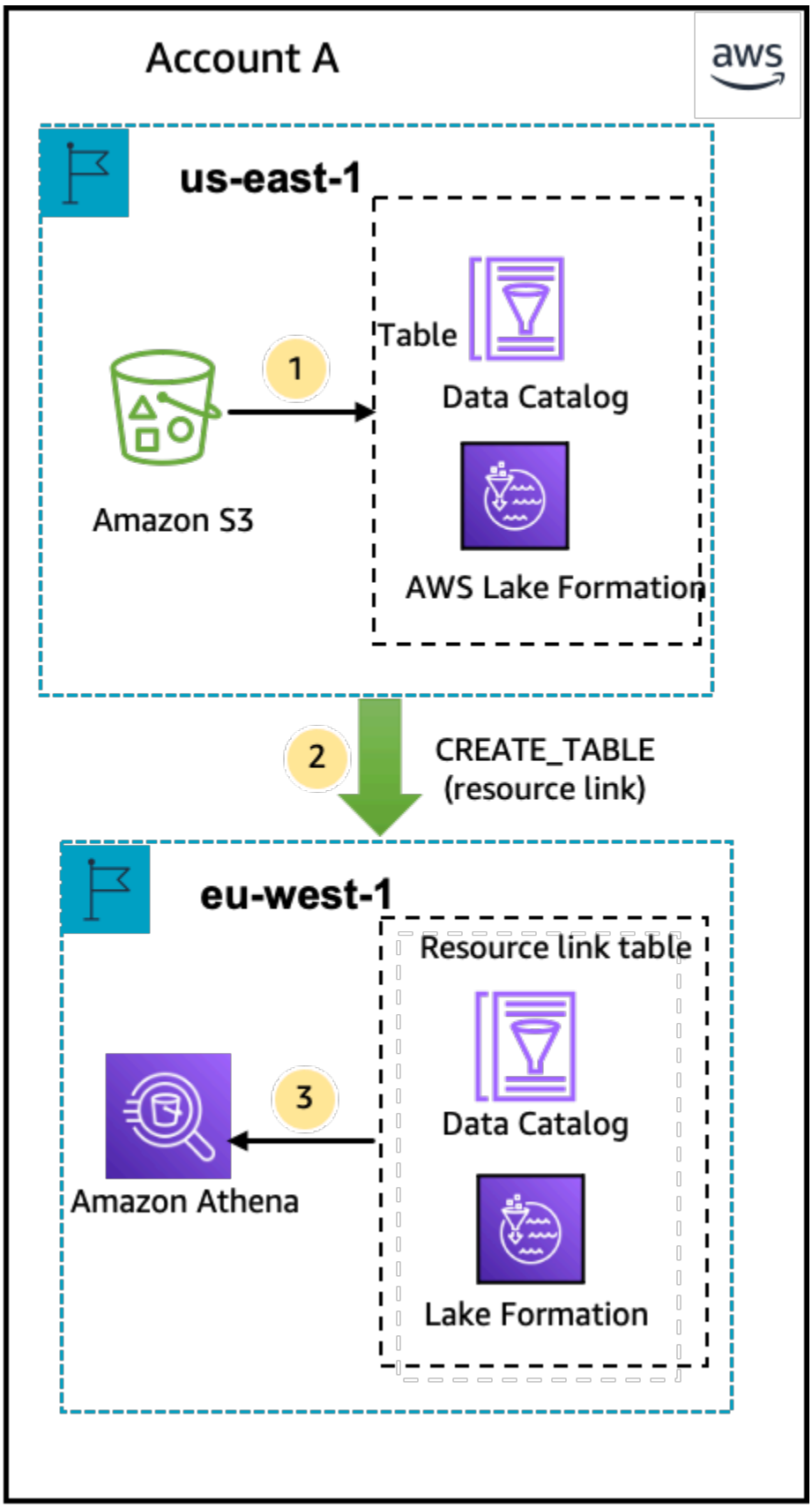
制限事項については、「クロスリージョンのデータアクセスに関する制限」を参照してください。

ワークフロー

次の図は、同じ AWS アカウントと外部アカウントから AWS リージョン間でデータにアクセスするためのワークフローを示しています。

同じ AWS アカウント内で共有されているテーブルにアクセスするためのワークフロー

次の図では、データは米国東部 (バージニア北部) リージョンの同じ AWS アカウントのユーザーと共有され、ユーザーは欧州 (アイルランド) リージョンから共有データをクエリします。


Account A
aws
us-east-1
Table
Data Catalog
1
Amazon S3
AWS Lake Formation
2
CREATE_TABLE
(resource link)
eu-west-1
Resource link table
Data Catalog
3
Amazon Athena
Lake Formation

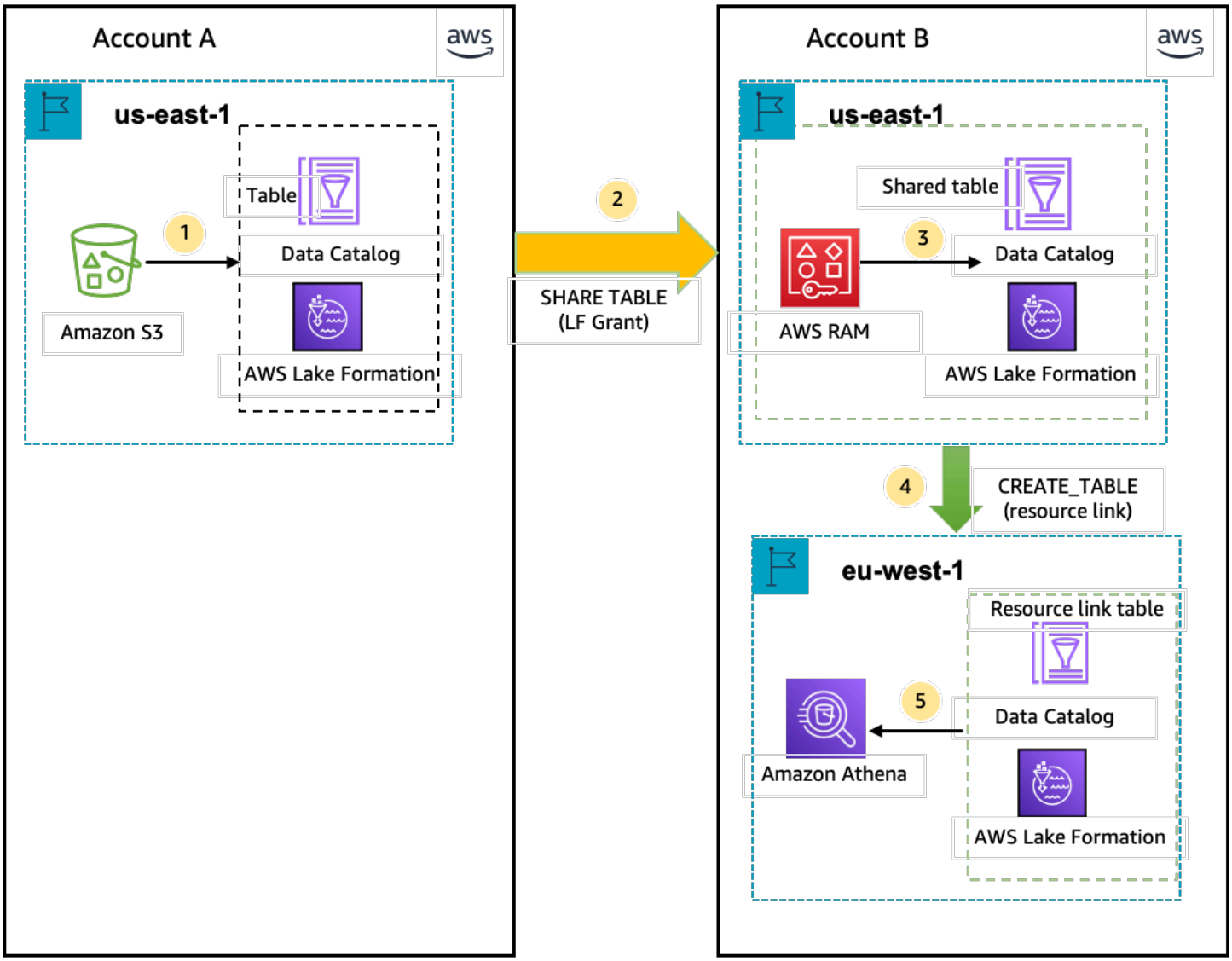
データレイク管理者は、以下のアクティビティ (ステップ 1～2) を実行します。

1. データレイク管理者は、Data Catalog データベースとテーブルで AWS アカウントを設定し、Amazon S3 データロケーションを米国東部 (バージニア北部) リージョンの Lake Formation に登録します。

   同じアカウントのプリンシパル (ユーザー) に対してデータカタログのリソース (図内の製品テーブル) への `Select` アクセス許可を付与します。
2. 米国東部 (バージニア北部) リージョンのソーステーブルを指すリソースリンクを欧州 (アイルランド) リージョンに作成します。欧州 (アイルランド) リージョンのリソースリンクへの `DESCRIBE` アクセス許可をプリンシパルに付与します。
3. ユーザーは Athena を使用して欧州 (アイルランド) リージョンからテーブルにクエリを実行します。

外部 AWS アカウントと共有されているテーブルにアクセスするためのワークフロー

下の図で、プロデューサーアカウント (アカウント A) は Amazon S3 バケットをホストし、データの場所を登録して、データカタログのテーブルを米国東部 (バージニア北部) リージョンのコンシューマーアカウント (アカウント B) と共有します。コンシューマーアカウント (アカウント B) のユーザーは、欧州 (アイルランド) リージョンからテーブルにクエリを実行します。



1. データレイク管理者は、米国東部 (バージニア北部) リージョンで、データカタログリソースと Lake Formation に登録された Amazon S3 データロケーションを使用して AWS アカウント (プロデューサーアカウント) を設定します。
2. プロデューサーアカウントのデータレイク管理者は、データカタログのテーブルをコンシューマーアカウントと共有します。
3. コンシューマーアカウントのデータレイク管理者は、米国東部 (バージニア北部) リージョンでデータ共有の招待を受け入れ、同じリージョンからプリンシパルに対して共有テーブルへの `Select` アクセス許可を付与します。
4. コンシューマーアカウントのデータレイク管理者は、米国東部 (バージニア北部) リージョンのターゲット共有テーブルを指すリソースリンクを欧州 (アイルランド) リージョンに作成し、欧州

(アイルランド) リージョンのリソースリンクへの DESCRIBE アクセス許可をユーザーに付与します。

5. ユーザーは Athena を使用して欧州 (アイルランド) リージョンからデータにクエリを実行します。

クロスリージョンのテーブルアクセスの設定

別のリージョンのデータにアクセスするには、まず Amazon S3 データの場所を登録したリージョンで、データカタログのデータベースとテーブルを設定する必要があります。データカタログのデータベースとテーブルは、自分のアカウントまたは別のアカウントのプリンシパルと共有できます。次に、ユーザーがデータにクエリを実行するリージョンで、ターゲット共有データの場所を指すリソースリンクを作成できるデータレイク管理者を作成する必要があります。

同じアカウント内の共有データに別のリージョンからクエリを実行するには

このセクションでは、ターゲット共有テーブルがあるリージョンをリージョン A とし、ユーザーはリージョン B からクエリを実行するものとします。

1. リージョン A (データを作成および共有する場所) でのアカウント設定

   データレイク管理者は、以下のアクションを実行する必要があります。

   a. Amazon S3 データの場所を登録します。

      詳細については、「データレイクへの Amazon S3 ロケーションの追加」を参照してください。

   b. アカウントでデータベースとテーブルを作成します。管理者以外のユーザーでも、データベースとテーブルを作成するアクセス許可があれば、作成できます。

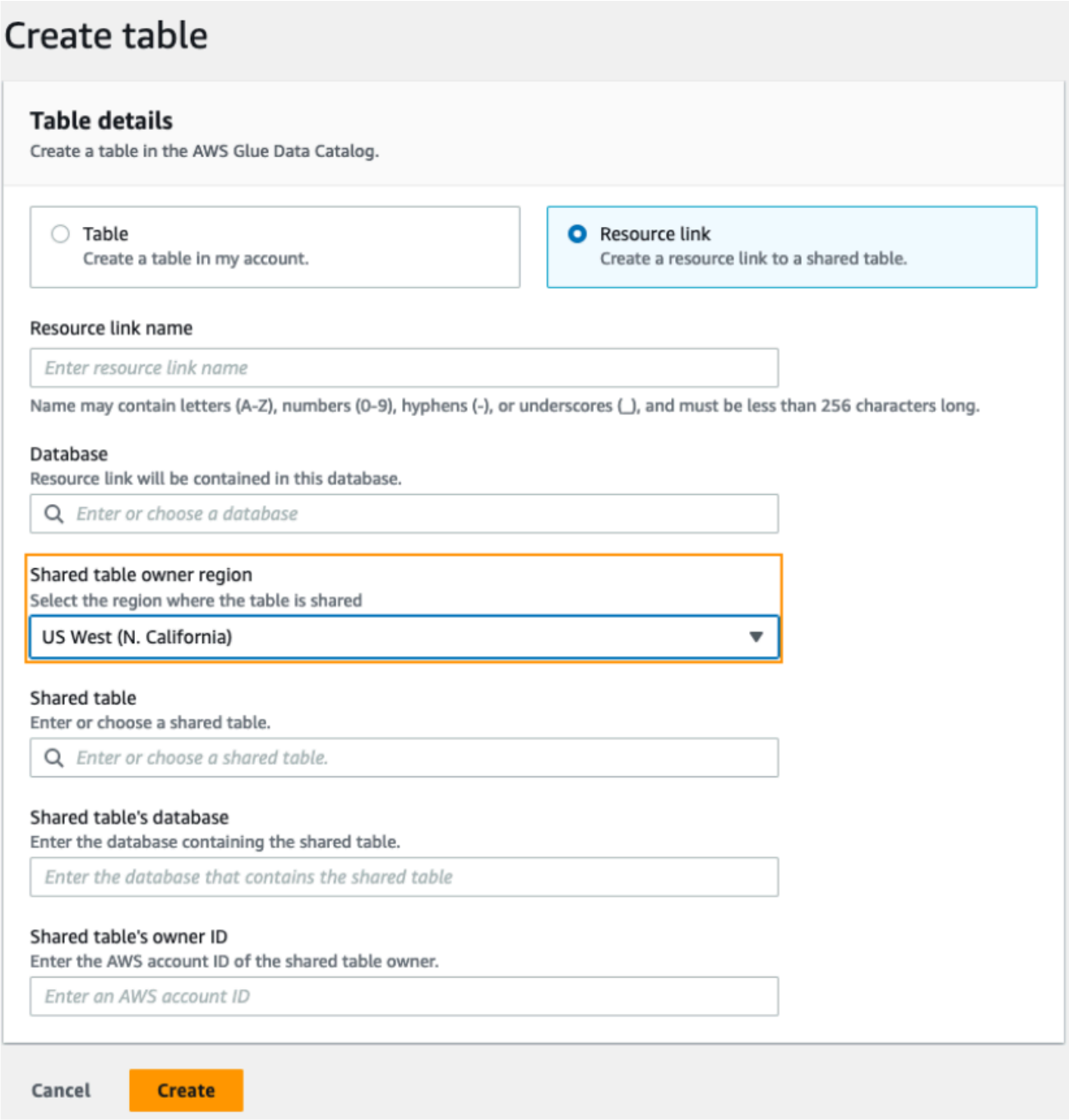
   c. `Grantable permissions` を使用して、テーブルのデータへのアクセス許可をプリンシパルに付与します。

      詳細については、「データカタログリソースに対するアクセス許可の付与」を参照してください。

2. リージョン B (データにアクセスする場所) でのアカウント設定

   データレイク管理者は、以下のアクションを実行する必要があります。

   a. リージョン A のターゲット共有テーブルを指すリソースリンクをリージョン B に作成します。[テーブルを作成] 画面で、[共有テーブル所有者のリージョン] を指定します。



データベースやテーブルへのリソースリンクの作成手順については、「リソースリンクの作成」を参照してください。

b. リージョン B のリソースリンクへの `Describe` アクセス許可を IAM プリンシパルに付与します。

リソースリンクへのアクセス許可の付与の詳細については、「リソースリンク許可の付与」を参照してください。

リージョン B の IAM プリンシパルは、Athena を使用してリンク経由でターゲットテーブルにクエリを実行できます。

別のリージョンのクロスアカウントデータにアクセスするには

1. プロデューサー/付与者のアカウント設定

   データレイク管理者は、以下のアクションを実行する必要があります。

   a. リージョン A でプロデューサー/付与者アカウントを設定します。
   b. Amazon S3 データの場所をリージョン A に登録します。
   c. データベースとテーブルを作成します。管理者以外のユーザーでも、テーブルを作成するアクセス許可があれば、作成できます。
   d. `Grantable permissions` を使用して、リージョン A のテーブルのデータへのアクセス許可をコンシューマー/被付与者アカウントに付与します。

      詳細については、「外部アカウントから AWS アカウント または IAM プリンシパル間で Data Catalog テーブルとデータベースを共有する」を参照してください。

2. コンシューマー/被付与者のアカウント設定

   データレイク管理者は、以下のアクションを実行する必要があります。

   a. リージョン A の からリソース共有の招待を受け入れ AWS RAM ます。
   b. 共有テーブルへのリソースリンクをリージョン B に作成します。リージョン B は、ユーザーがテーブルにクエリを実行する場所です。
   c. リージョン A で共有テーブルのデータへのアクセス許可を IAM プリンシパルに付与します。

      > **Note**
      >
      > テーブルを共有したのと同じリージョンで、共有したテーブルにアクセス許可を付与する必要があります。

   d. リージョン B でリソースリンクへのアクセス許可をプリンシパルに付与します。

      リージョン B のコンシューマーアカウントのプリンシパルは、Athena を使用してリージョン B から共有テーブルにクエリを実行します。

AWS Lake Formation のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。

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して説明しています。

- 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。「AWS 」 コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。 AWS Lake Formation に適用されるコンプライアンスプログラムの詳細については、AWS「コンプライアンスプログラムによる対象範囲内のサービス」を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

本書は、Lake Formation の使用時に責任共有モデルを適用する方法を理解するために役立ちます。以下のトピックでは、セキュリティとコンプライアンスの目的を達成するために Lake Formation を設定する方法が紹介されています。また、Lake Formation リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

トピック

- Lake Formation におけるデータ保護
- AWS Lake Formation のインフラストラクチャセキュリティ
- サービス間の混乱した代理の防止
- AWS Lake Formation のセキュリティイベントログ記録

Lake Formation におけるデータ保護

AWS の責任共有モデル は、データ保護に適用されます。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、「Data Privacy FAQChina」を参照してください。 欧州におけるデータ保護に関する情報については、General Data Protection Regulation (GDPR) Center 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、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。

保管時の暗号化

AWS Lake Formation は、以下の領域でデータ暗号化をサポートしています。

- Amazon Simple Storage Service (Amazon S3) データレイク内のデータ。

Lake Formation は、AWS Key Management Service (AWS KMS) を使用したデータの暗号化をサポートします。データは通常、AWS Glue の抽出、変換、ロード (ETL) ジョブを用いてデータレイクに書き込まれます。AWS Glue ジョブによって書き込まれたデータを暗号化する方法については、「AWS Glue デベロッパーガイド」の「クローラー、ジョブ、および開発エンドポイントによって書き込まれたデータの暗号化」を参照してください。

- は AWS Glue Data Catalog、Lake Formation がデータレイク内のデータを記述するメタデータテーブルを保存する場所です。

  詳細については、「AWS Glue デベロッパーガイド」の「データカタログの暗号化」を参照してください。

Amazon S3 ロケーションをデータレイクのストレージとして追加するには、そのロケーションを AWS Lake Formation に登録します。その後、このロケーションをポイントする AWS Glue Data Catalog オブジェクトと、そのロケーション内の基盤となるデータに対する細粒度のアクセスコントロールのために Lake Formation 許可を使用することができます。

Lake Formation は、暗号化されたデータが含まれる Amazon S3 ロケーションの登録をサポートします。詳細については、「暗号化された Amazon S3 ロケーションの登録」を参照してください。

AWS Lake Formation のインフラストラクチャセキュリティ

マネージドサービスである AWS Lake Formation は、ホワイトペーパー「Amazon Web Services: セキュリティプロセスの概要」に記載されている AWS グローバルネットワークセキュリティ手順で保護されています。

AWS 公開された API コールを使用して、ネットワーク経由で Lake Formation にアクセスします。クライアントで Transport Layer Security (TLS) 1.0 以降がサポートされている必要があります。TLS 1.2 以降が推奨されています。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストにはアクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

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ためのアクセス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の実行を強制できてしまう場合に生じる、セキュリティ上の問題です。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。これを防ぐために、 は、アカウント内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を持つすべてのサービスのデータを保護するのに役立つツール AWS を提供します。

リソースポリシーで aws:SourceArn および aws:SourceAccount のグローバル条件コンテキストキーを使用して、AWS Lake Formation が別のサービスに付与する許可をそのリソースに制限することをお勧めします。両方のグローバル条件コンテキストキーを使用しており、それらが同じポリシーステートメントで使用されるときは、aws:SourceAccount 値と、aws:SourceArn 値のアカウントが同じアカウント ID を使用する必要があります。

現在、Lake Formation は以下の形式の aws:SourceArn のみをサポートしています。

```
arn:aws:lakeformation:aws-region:account-id:*
```

以下は、混乱した代理問題を防ぐために Lake Formation で aws:SourceArn および aws:SourceAccount のグローバル条件コンテキストキーを使用する方法を示す例です。

JSON

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

```
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:lakeformation:aws-region:123456789012:*"
        }
      }
    }
  ]
}
```

AWS Lake Formation のセキュリティイベントログ記録

AWS Lake Formation は AWS CloudTrail、Lake Formation のユーザー、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、Lake Formation のすべての API コールをイベントとして取得します。キャプチャされた呼び出しには、Lake Formation コンソールからの呼び出し、 AWS Command Line Interface、Lake Formation API オペレーションへのコード呼び出しが含まれます。

Lake Formation でのイベントロギングに関する詳細については、「を使用した AWS Lake Formation API コールのログ記録 AWS CloudTrail」を参照してください。

> **Note**
>
> `GetTableObjects`、`UpdateTableObjects`、および `GetWorkUnitResults` は、大容量のデータプレーン操作です。これらの API に対する呼び出しは、現在 CloudTrail にはログされません。CloudTrail でのデータプレーンオペレーションに関する詳細については、「AWS CloudTrail ユーザーガイド」の「証跡のデータイベントをログ記録する」を参照してください。
>
> 追加の CloudTrail イベントをサポートするための Lake Formation での変更は、「のドキュメント履歴 AWS Lake Formation」に掲載されます。

サードパーティーサービスと Lake Formation との統合

AWS Lake Formation と統合することで、サードパーティーサービスは Amazon S3 ベースのデータレイクにあるデータに安全にアクセスできます。Lake Formation を認可エンジンとして使用して、AWS Glue ETL、Amazon Athena、Amazon EMR、Redshift Spectrum などの統合 AWS サービスでデータレイクへのアクセス許可を管理または適用できます。Lake Formation には、サービスを統合するための 2 つのオプションがあります。

1. Lake Formation アプリケーション統合設定: Lake Formation は、有効なアクセス許可に基づいて、スコープダウンされた一時的な認証情報を AWS STS トークンの形式で登録された Amazon S3 ロケーションに供給できるため、承認されたアプリケーションはユーザーに代わってデータにアクセスできます。
2. 一元的な適用: Lake Formationのクエリ API 操作は、Amazon S3 からデータを取得し、有効な許可に基づいて結果をフィルタリングします。クエリ API 操作と統合するエンジンやアプリケーションは、呼び出し元のアイデンティティの許可を評価し、これらの許可に基づいてデータをセキュアにフィルタリングする作業を Lake Formation に依存できます。サードパーティークエリエンジンは、フィルタリングされたデータのみを認識して操作します。

Lake Formation の認証情報供給は、Spark SQL クエリと統合されません。認証情報供給は、ETL AWS Glue ライブラリを介して実行されるクエリでのみ機能します。

トピック

- Lake Formation アプリケーション統合の使用

Lake Formation アプリケーション統合の使用

Lake Formationを使用すると、サードパーティのサービスを Lake Formation と統合して、ユーザーに代わって Amazon S3 データに一時的にアクセスできるようになります。これには、GetTemporaryGlueTableCredentials 操作と GetTemporaryGluePartitionCredentials 操作を使用します。これにより、サードパーティサービスは他の AWS 分析サービスが使用するのと同じ認可および認証情報供給機能を使用できます。このセクションでは、これらの API 操作を使用してサードパーティクエリエンジンを Lake Formation と統合する方法について説明します。

これらの API 操作はデフォルトでは無効になっています。Lake Formation にアプリケーションの統合を許可するには、次の 2 つのオプションがあります。

- アプリケーション統合 API 操作が呼び出されるたびに検証される IAM セッションタグを設定する

  詳細については、「サードパーティのクエリエンジンがアプリケーション統合 API 操作を呼び出すアクセス許可を有効にする」を参照してください。

- [外部エンジンが Amazon S3 ロケーションのデータにフルテーブルアクセスでアクセスするのを許可する] オプションを有効にする

  このオプションにより、ユーザーがフルテーブルアクセス権を持っている場合、クエリエンジンとアプリケーションは IAM セッションタグなしで認証情報を取得できます。クエリエンジンとアプリケーションのパフォーマンスが向上し、データアクセスが簡単になります。Amazon EC2 での Amazon EMR では、この設定を活用できます。

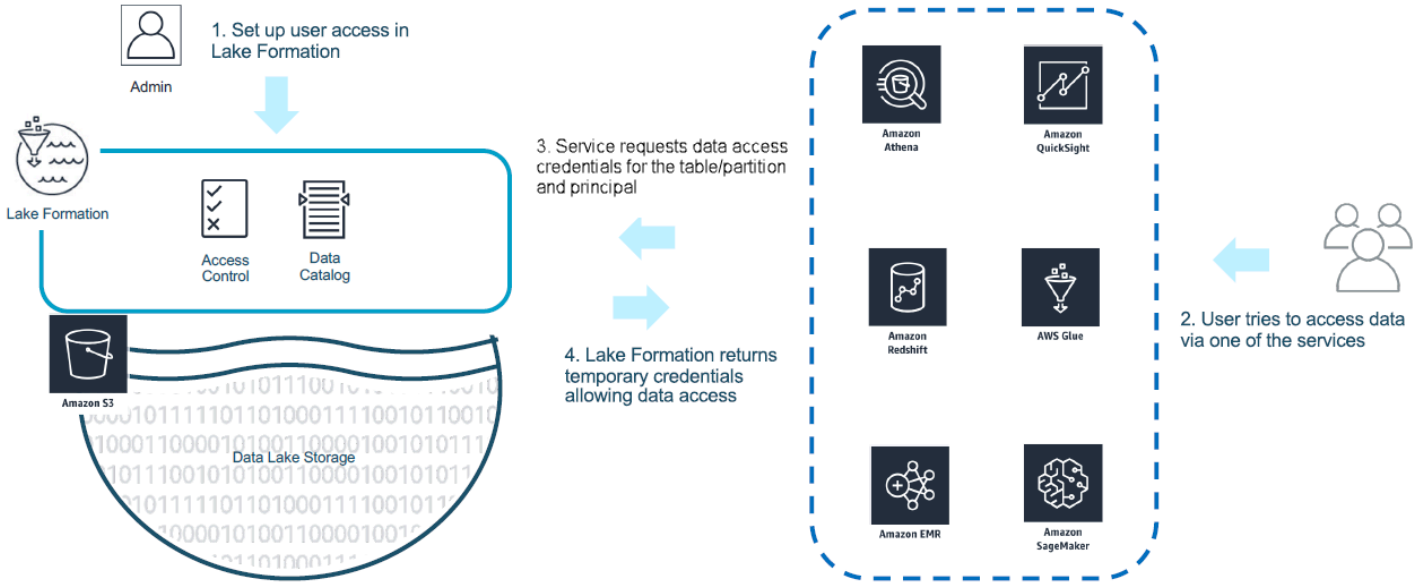
  詳細については、「フルテーブルアクセスのためのアプリケーション統合」を参照してください。

トピック

- Lake Formation アプリケーション統合の仕組み
- Lake Formation アプリケーション統合におけるロールと責任
- アプリケーション統合 API 操作の Lake Formation ワークフロー
- サードパーティークエリエンジンの登録
- サードパーティのクエリエンジンがアプリケーション統合 API 操作を呼び出すアクセス許可を有効にする
- フルテーブルアクセスのためのアプリケーション統合

Lake Formation アプリケーション統合の仕組み

このセクションでは、アプリケーション統合 API 操作を使用してサードパーティアプリケーション (クエリエンジン) を Lake Formation と統合する方法について説明します。



1. Lake Formation 管理者が以下のアクティビティを実行します。
   - Amazon S3 ロケーション内のデータにアクセスするための適切な許可を持つ IAM ロール (認証情報の供給用) を提供することで、Amazon S3 ロケーションを Lake Formation に登録する。
   - Lake Formation の認証情報供給 API 操作を呼び出すことができるようにサードパーティーアプリケーションを登録する。「the section called “サードパーティークエリエンジンの登録”」を参照してください。
   - データベースとテーブルにアクセスするための許可をユーザーに付与する。

     例えば、個人を特定できる情報 (PII) を示す複数の列が含まれているユーザーセッションデータセットを公開する場合、アクセスを制限するには、これらの列に LF-TBAC タグを「classification」という名前および「sensitive」という値で割り当てます。次に、ユーザーセッションデータへのアクセス権をビジネスアナリストに付与するための許可を定義します。ただし、classification = sensitive がタグ付けされた列は除きます。
2. プリンシパル (ユーザー) が、統合されたサービスにクエリを送信します。
3. 統合されたアプリケーションが Lake Formation にリクエストを送信し、テーブル情報とテーブルにアクセスするための認証情報を要求します。
4. クエリを実行するプリンシパルにテーブルへのアクセスが認可されている場合は、Lake Formation から統合されたアプリケーションに認証情報を返し、データアクセスを許可します。

> **Note**
>
> Lake Formation は、認証情報を提供するときに基盤のデータにアクセスしません。

5. 統合されたサービスが Amazon S3 からデータを読み取り、受け取ったポリシーに基づいて列をフィルタリングして、結果をプリンシパルに返します。

> **Important**
>
> Lake Formation の認証情報供給 API 操作は、失敗時における明示的な拒否 (フェイルクローズ) モデルを使用した分散型適用を有効にします。これにより、顧客、サードパーティーサービス、Lake Formation の間に 3 パーティーセキュリティモデルが導入されます。統合されたサービスでは、Lake Formation 許可が適切に適用 (分散型適用) されます。

統合されたサービスは、Lake Formation から返されたポリシーに基づいて Amazon S3 からデータを読み取ってフィルタリングし、フィルタリングしたデータをユーザーに返す責任があります。統合されたサービスは、フェイルクローズモデルに従います。つまり、適切な Lake Formation 許可を適用できない場合はクエリを失敗させる必要があります。

Lake Formation アプリケーション統合におけるロールと責任

以下は、 とのサードパーティーアプリケーション統合を有効にするためのロールとそれに関連する責任です AWS Lake Formation。

ロール	責任
顧客	• Lake Formation アプリケーション統合設定を有効にする (「the section called “サードパーティークエリエンジンの登録”」を参照)。 • 承認されたサードパーティーを Lake Formation に明示的に登録する (「the section called “サードパーティークエリエンジンの登録”」を参照) • Lake Formation の許可でサードパーティーソリューションをテストして検証する • サードパーティーによる Lake Formation 認証情報供給 API 操作の使用状況をモニタリングおよび監査する

ロール	責任
サードパーティー	• 各ソフトウェアリビジョンでサポートされる機能を公に文書化し、その機能を正しく有効化する手順を提供する • Lake Formation の認証情報供給 API 操作を (ドキュメントに従って) 呼び出すときにサポートされる機能を正確にアドバタイズする • 供給された認証情報を安全に保管して処理し、認証情報の漏洩や権限昇格を回避する • サポートされている機能に基づいて許可を適用し、フィルタリングしたデータのみをユーザーに返す • 必要な許可を適切に適用できない場合はクエリを失敗させる
AWS Lake Formation	• 該当するプリンシパルに対する有効な許可を正しく取得して返す • API 操作のコールごとにサードパーティーがサポートする機能を検証する • エンジンのアドバタイズされた機能がカタログリソースで定義されている機能と一致する場合にのみ、スコープダウンされた IAM 認証情報を返し、一致しない場合はエラーを返す

アプリケーション統合 API 操作の Lake Formation ワークフロー

アプリケーション統合 API 操作のワークフローは次のとおりです。

1. ユーザーが、統合されたサードパーティークエリエンジンを使用してデータへのクエリまたはリクエストを送信します。クエリエンジンがユーザーまたはユーザーのグループを表す IAM ロールを引き受けて、信頼できる認証情報を取得し、これを使用してアプリケーション統合 API 操作を呼び出します。
2. クエリエンジンが GetUnfilteredTableMetadata (パーティション化されたテーブルの場合は GetUnfilteredPartitionsMetadata) を呼び出し、Data Catalog からメタデータとポリシー情報を取得します。
3. Lake Formation がリクエストの認可を実行します。ユーザーがテーブルに対する適切な許可を持っていない場合は、AccessDeniedException がスローされます。
4. リクエストの一部として、クエリエンジンが、サポートするフィルタリングを送信します。配列内で送信できるフラグには、COLUMN_PERMISSIONS と CELL_FILTER_PERMISSION の 2 つがあります。クエリエンジンがこれらの機能のどちらもサポートしておらず、機能に関するポリ

シーがテーブルに存在する場合は、PermissionTypeMismatchException がスローされ、クエリが失敗します。これは、データ漏洩を防ぐためのものです。

5. 返される応答には以下が含まれます。
   - テーブルの完全なスキーマ。クエリエンジンがこれを使用してストレージからのデータを解析できるようにするためです。
   - ユーザーがアクセスできる認可された列のリスト。認可された列のリストが空の場合は、ユーザーに DESCRIBE 許可があっても SELECT 許可がないことを示し、クエリが失敗します。
   - IsRegisteredWithLakeFormation というフラグ。これは、Lake Formation がこのリソースデータに認証情報を供給できるかどうかを示します。これが false を返す場合、Amazon S3 へのアクセスには顧客の認証情報を使用する必要があります。
   - データの行に適用する必要がある CellFilters のリスト (存在する場合)。このリストには、列と、各行を評価する式が含まれています。これは、CELL_FILTER_PERMISSION をリクエストの一部として送信し、テーブルに対するデータフィルターが呼び出し側のユーザーにある場合にのみ入力されます。
6. メタデータを取得すると、クエリエンジンは GetTemporaryGlueTableCredentialsまたは GetTemporaryGluePartitionCredentialsを呼び出して、Amazon S3 の場所からデータを取得するための AWS 認証情報を取得します。
7. クエリエンジンが Amazon S3 から関連するオブジェクトを読み取り、ステップ 2 で受け取ったポリシーに基づいてデータをフィルタリングして、ユーザーに結果を返します。

Lake Formation のアプリケーション統合 API 操作には、サードパーティクエリエンジンとの統合を設定するための追加のコンテンツが含まれています。操作の詳細については、「認証情報供給 API 操作」セクションを参照してください。

QuerySessionContext は、これらのアプリケーション統合 API オペレーションLake Formationのためにクエリエンジンが追加で に送信できる構造です。これにより、 Lake Formation は特定のクエリに追加のコンテキストを保存して使用できます。QuerySessionContext の使用方法の例を次に示します。

1. クエリエンジンはGetInternalUnfilteredMetadata呼び出しを行い、リクエストに一意のクエリ ID を含む QSC 構造を渡します。

```
{
    "QuerySessionContext": {
        "QueryId": "your-unique-identifier-here"
    }
```

```
}
```

2. GetInternalUnfilteredMetadata 呼び出しはレスポンスでQueryAuthorizationId文字列を返します。入力で QSC 構造を受け入れる次の (およびそれ以降の) クエリ呼び出しでは、クエリエンジンは によってQueryAuthorizationId返される も含む同じ QSC 構造を渡しますLake Formation。次の呼び出しが であるとしますGetTemporaryGlueTableCredentials。リクエストには以下が含まれます。

```
{
    "QuerySessionContext": {
        "QueryAuthorizationId": "lf-returned-query-authz-id-here",
        "QueryId": "your-unique-identifier-here"
    },
}
```

サードパーティークエリエンジンの登録

サードパーティクエリエンジンがアプリケーション統合 API 操作を使用するには、クエリエンジンがユーザーに代わって API 操作を呼び出すアクセス許可を明示的に有効にする必要があります。これを行うには、以下のステップに従います。

1. Lake Formation コンソール、、 AWS CLI または API/SDK を介して AWS アプリケーション統合 API オペレーションを呼び出すアクセス許可が必要な AWS アカウントと IAM セッションタグを指定する必要があります。
2. サードパーティーのクエリエンジンがアカウントで実行ロールを引き受ける場合、クエリエンジンは、Lake Formation に登録されている、サードパーティーエンジンを表すセッションタグをアタッチする必要があります。Lake Formation は、このタグを使用して、承認されたエンジンからのリクエストであるかどうかを検証します。セッションタグの詳細については、「IAM ユーザーガイド」のセッションタグに関するセクションを参照してください。
3. サードパーティクエリエンジンの実行ロールを設定する場合は、IAM ポリシーに少なくとも以下の一連の許可が必要です。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": {"Effect": "Allow",
    "Action": [
      "lakeformation:GetDataAccess",
```

```
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:CreateDatabase",
        "glue:GetUserDefinedFunction",
        "glue:GetUserDefinedFunctions",
        "glue:GetPartition",
        "glue:GetPartitions"
      ],
      "Resource": "*"
    }
}
```

4. クエリエンジンの実行ロールにロール信頼ポリシーを設定して、このロールにどのセッションタグのキーバリューペアをアタッチできるかを細かくアクセス制御します。次の例で、このロールはセッションタグのキーとして "LakeFormationAuthorizedCaller"、セッションタグのバリューとして "engine1" をアタッチすることのみが許可され、他のセッションタグのキーバリューペアは許可されません。

```
{
    "Sid": "AllowPassSessionTags",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/query-execution-role"
    },
    "Action": "sts:TagSession",
    "Condition": {
    "StringLike": {
        "aws:RequestTag/LakeFormationAuthorizedCaller": "engine1"        }
    }
}
```

LakeFormationAuthorizedCaller が、STS:AssumeRole API 操作を呼び出してクエリエンジンが使用する認証情報を取得する場合は、セッションタグを AssumeRole リクエストに含める必要があります。返された一時的な認証情報を使用して、Lake Formation アプリケーション統合 API リクエストを実行することができます。

Lake Formation アプリケーション統合 API 操作では、呼び出し元プリンシパルが IAM ロールである必要があります。この IAM ロールには、Lake Formation に登録されている定義済みの値を持つセッションタグを含める必要があります。このタグにより、Lake Formation は、アプリケーション統合

API 操作の呼び出しに使用されたロールが、この呼び出しを行う許可を得ていることを確認できます。

サードパーティのクエリエンジンがアプリケーション統合 API 操作を呼び出すアクセス許可を有効にする

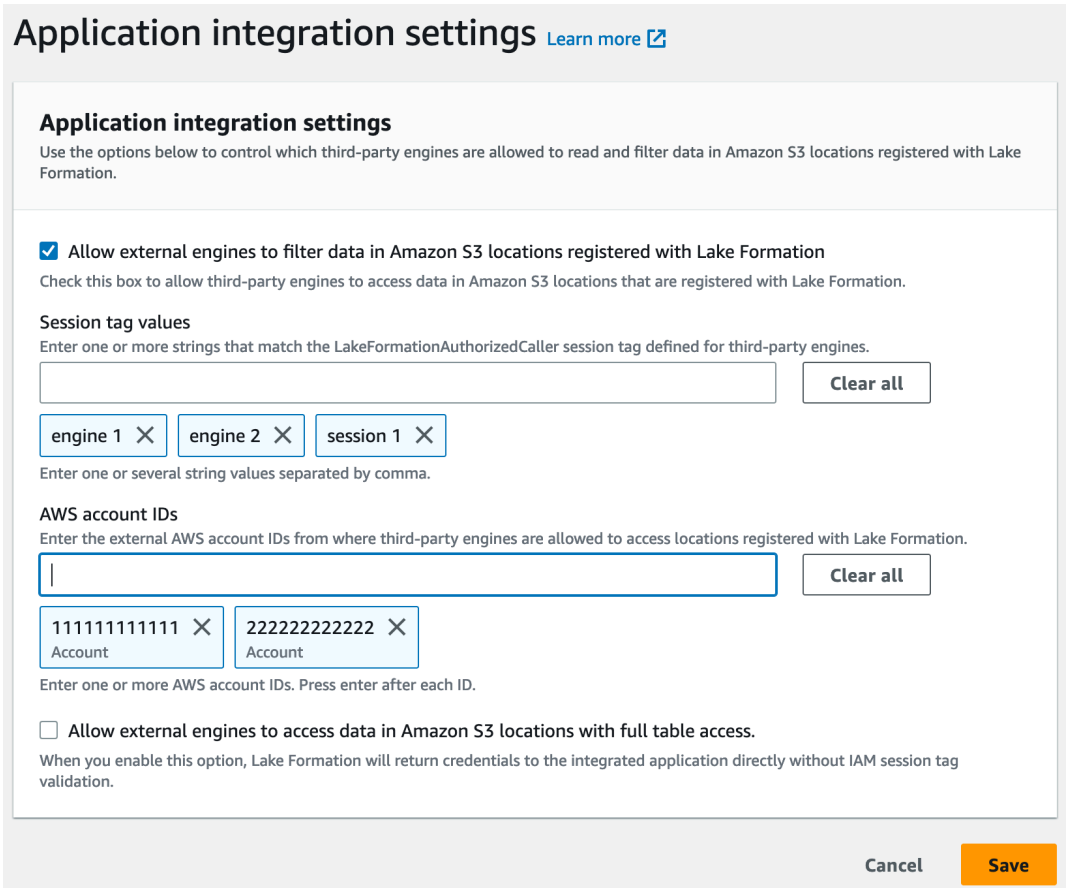
サードパーティーのクエリエンジンがコンソール、、 AWS CLI または API/SDK を介してアプリケーション統合 API オペレーションを AWS Lake Formation呼び出すことを許可するには、次の手順に従います。

Console

外部データフィルタリングのためのアカウントを登録するには

1. にサインインし AWS マネジメントコンソール、https://console.aws.amazon.com/lakeformation/ で Lake Formation コンソールを開きます。
2. 左側のナビゲーションで、[管理] を展開し、[アプリケーション統合設定] を選択します。
3. [アプリケーション統合設定] ページで、[外部エンジンが Lake Formation に登録された Amazon S3 ロケーション内のデータをフィルタリングすることを許可する] オプションを選択します。
4. サードパーティエンジン用に作成したセッションタグを入力します。セッションタグの詳細については、AWS Identity and Access Management 「 ユーザーガイド」の AWS 「STS でのセッションタグの受け渡し」を参照してください。
5. サードパーティーエンジンを使用して現在のアカウントにあるリソースのフィルタリングされていないメタデータ情報やデータアクセス認証情報にアクセスできるユーザーのアカウント ID を入力します。

   AWS アカウント ID フィールドを使用して、クロスアカウントアクセスを設定することもできます。



CLI

`put-data-lake-settings` CLI コマンドを使用して以下のパラメータを設定します。

この AWS CLI コマンドを使用する際に設定するフィールドは 3 つあります。

- `allow-external-data-filtering` – (ブール) サードパーティーエンジンが、現在のアカウントにあるリソースのフィルタリングされていないメタデータ情報とデータアクセス認証情報にアクセスできることを示します。
- `external-data-filtering-allow-list` – (配列) サードパーティーエンジンの使用時に、現在のアカウントにあるリソースのフィルタリングされていないメタデータ情報とデータアクセス認証情報にアクセスできるアカウント ID のリストです。AllowExternalDataFiltering が true に設定されている場合、ExternalDataFilteringAllowList プロパティには少なくとも 1 つのアカウント ID を含める必要があります。空のリストは使用できません。
- `authorized-sessions-tag-value-list` – (配列) 認可されたセッションタグ値 (文字列) のリストです。IAM ロールの認証情報に認可されたキーバリューペアがアタッチされている場合、セッションタグがリストに含まれていると、現在のアカウントにあ

るリソースのフィルタリングされていないメタデータ情報とデータアクセス認証情報に対するアクセス権がセッションに付与されます。認可されたセッションタグキーは *LakeFormationAuthorizedCaller* として定義されます。

- AllowFullTableExternalDataAccess - (ブール値) 呼び出し元が完全なデータアクセスアクセス許可を持っている場合に、サードパーティのクエリエンジンがセッションタグなしでデータアクセス認証情報を取得することを許可するかどうか。

例えば、次のようになります。

```
aws lakeformation put-data-lake-settings --cli-input-json file://
datalakesettings.json

{
  "DataLakeSettings": {
    "DataLakeAdmins": [
      {
        "DataLakePrincipalIdentifier": "arn:aws:iam::111111111111:user/lakeAdmin"
      }
    ],
    "CreateDatabaseDefaultPermissions": [],
    "CreateTableDefaultPermissions": [],
    "TrustedResourceOwners": [],
    "AllowExternalDataFiltering": true,
    "ExternalDataFilteringAllowList": [
        {"DataLakePrincipalIdentifier": "111111111111"}
        ],
    "AuthorizedSessionTagValueList": ["engine1"],
    "AllowFullTableExternalDataAccess": false
    }


}
```

API/SDK

PutDataLakeSetting API 操作を使用して以下のパラメータを設定します。

この API 操作を使用する場合は、以下の 3 つのフィールドを設定します。

- AllowExternalDataFiltering – (ブール) サードパーティーエンジンが、現在のアカウントにあるリソースのフィルタリングされていないメタデータ情報とデータアクセス認証情報にアクセスできるかどうかを示します。
- ExternalDataFilteringAllowList – (配列) サードパーティーエンジンを使用して、現在のアカウントにあるリソースのフィルタリングされていないメタデータ情報とデータアクセス認証情報にアクセスできるアカウント ID のリストです。
- AuthorizedSectionsTagValueList – (配列) 認可されたタグ値 (文字列) のリストです。IAM ロールの認証情報に認可済みのタグがアタッチされている場合は、設定されたアカウントにあるリソースのフィルタリングされていないメタデータ情報とデータアクセス認証情報に対するアクセス権がセッションに付与されます。認可済みのセッションタグキーは *LakeFormationAuthorizedCaller* として定義されます。
- AllowFullTableExternalDataAccess - (ブール値) 呼び出し元が完全なデータアクセスアクセス許可を持っている場合に、サードパーティのクエリエンジンがセッションタグなしでデータアクセス認証情報を取得することを許可するかどうか。

例えば、次のようになります。

```
//Enable session tag on existing data lake settings
public void sessionTagSetUpForExternalFiltering(AWSLakeFormationClient
 lakeformation) {
    GetDataLakeSettingsResult getDataLakeSettingsResult =
 lfClient.getDataLakeSettings(new GetDataLakeSettingsRequest());
    DataLakeSettings dataLakeSettings =
 getDataLakeSettingsResult.getDataLakeSettings();

    //set account level flag to allow external filtering
    dataLakeSettings.setAllowExternalDataFiltering(true);

    //set account that are allowed to call credential vending or Glue
 GetFilteredMetadata API
    List<DataLakePrincipal> allowlist = new ArrayList<>();
    allowlist.add(new
 DataLakePrincipal().withDataLakePrincipalIdentifier("111111111111"));
    dataLakeSettings.setWhitelistedForExternalDataFiltering(allowlist);

    //set registered session tag values
    List<String> registeredTagValues = new ArrayList<>();
    registeredTagValues.add("engine1");
    dataLakeSettings.setAuthorizedSessionTagValueList(registeredTagValues);
```

```
    lakeformation.putDataLakeSettings(new
 PutDataLakeSettingsRequest().withDataLakeSettings(dataLakeSettings));
}
```

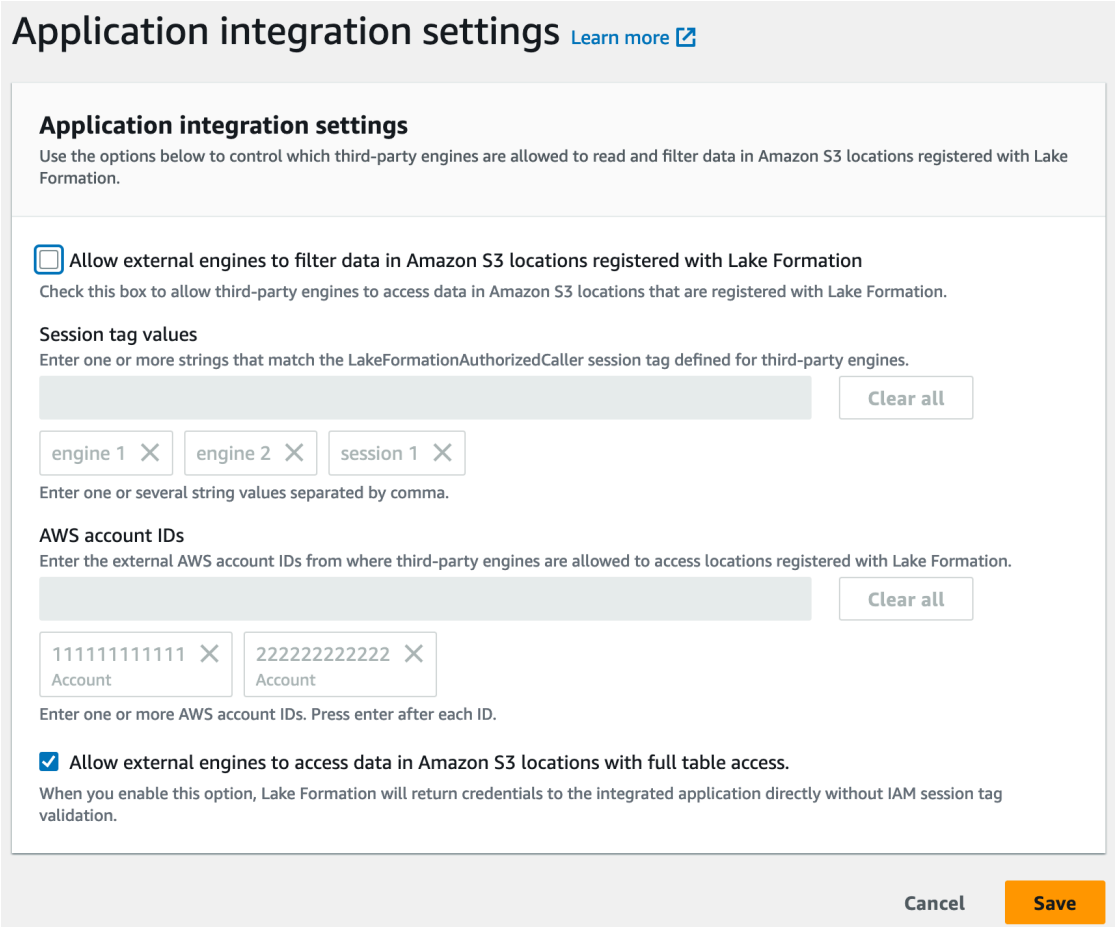
フルテーブルアクセスのためのアプリケーション統合

以下の手順に従って、サードパーティのクエリエンジンが IAM セッションタグの検証なしでデータにアクセスできるようにします。

Console

1. Lake Formation コンソール (https://console.aws.amazon.com/lakeformation/) にサインインします。
2. 左側のナビゲーションで、[管理] を展開し、[アプリケーション統合設定] を選択します。
3. [アプリケーション統合設定] ページで、[外部エンジンがフルテーブルアクセスで Amazon S3 ロケーション内のデータにアクセスすることを許可する] オプションを選択します。

   このオプションを有効にすると、Lake Formation は IAM セッションタグの検証なしに、認証情報をクエリ元のアプリケーションに直接返します。



AWS CLI

`put-data-lake-settings` CLI コマンドを使用して、`AllowFullTableExternalDataAccess` パラメータを設定します。

```
aws lakeformation put-data-lake-settings –cli-input-json file://put-data-lake-
settings.json –region ap-northeast-1
{
    "DataLakeSettings": {
        "DataLakeAdmins": [
            {
                "DataLakePrincipalIdentifier": "arn:aws:iam::111111111111:user/
lakeAdmin"
            }
        ],
        "AllowFullTableExternalDataAccess": true
    }
}
```

他の AWS サービスの使用

AWS Amazon Athena、Amazon Redshift Spectrum AWS Glue、Amazon EMR などの のサービスでは、 を使用して、Lake Formation に登録されている Amazon S3 ロケーションのデータ AWS Lake Formation に安全にアクセスできます。Lake Formation では、 AWS Glue Data Catalog内のテーブルに対して、きめ細かいアクセスコントロール (FGAC) のアクセス許可を定義して管理できます。これらの各 AWS サービスは Lake Formation の信頼できる発信者であり、Lake Formation は一時的な認証情報を通じて Amazon S3 に保存されているデータへのアクセスを提供します。詳細については、「Lake Formation アプリケーション統合の仕組み」を参照してください。

これらの機能を利用するには、Lake Formation で最初に Amazon S3 ロケーションを登録し、テーブル、データベース、Amazon S3 ロケーションにアクセスするための適切なアクセス許可を IAM プリンシパルに割り当てる必要があります。詳細については、Lake Formation 許可の管理 を参照してください。

次の表は、Amazon Athena、、Amazon EMR AWS Glue、Amazon Redshift Spectrum が、Amazon S3 に保存されているデータとデータカタログのテーブルメタデータを使用して、 AWS Glue 標準テーブルとトランザクションテーブル (Apache Iceberg、Apache Hudi、Linux 基盤 Delta Lake) のデータにアクセスするためにサポートする Lake Formation アクセス許可のタイプを示しています。

AWS AWS Glue 標準テーブルとビューでサポートされている サービスとアクセス許可タイプ

AWS サービス	テーブルレベルのアクセス許可	列レベルのアクセス許可	行レベルおよびセルレベルのアクセス許可
Athena SQL	読み取り/書き込みアクセス	読み取りアクセス	読み取りアクセス
Athena Spark	サポートされません	サポートされません	サポートされません
プロビジョニングされたクラスターまたは Amazon Redshift Serverless 上の Redshift Spectrum	読み取り/書き込みアクセス	読み取りアクセス	読み取りアクセス

AWS サービス	テーブルレベルのアクセス許可	列レベルのアクセス許可	行レベルおよびセルレベルのアクセス許可
Amazon EMR (EC2) 上の Apache Spark	読み取り/書き込みアクセス	読み取りアクセス	読み取りアクセス
Amazon EMR (EC2) 上の Apache Hive	読み取り/書き込みアクセス	読み取りアクセス	サポートされていません
EMR Serverless 上の Apache Spark	読み取り/書き込みアクセス	読み取りアクセス	読み取りアクセス
EMR Serverless 上の Apache Hive	サポートされません	サポートされません	サポートされません
Amazon EMR on EKS	サポートされません	サポートされません	サポートされません
AWS Glue ETL	読み取り/書き込みアクセス	AWS Glue 5.0 以降では、読み取りアクセスがサポートされています。	AWS Glue 5.0 以降では、読み取りアクセスがサポートされています。

考慮事項と制限事項

- Athena Spark では、Lake Formation アクセス許可によるデータカタログテーブルのクエリはサポートされません。
- Athena SAML ベースのユーザーは、SAML 2.0 ベースのフェデレーションを有効にすることで、Lake Formation アクセス許可で保護されたデータソースを読み取ることができます。SAML ユーザーは Parquet テーブルにデータを挿入できます。
- EMR Serverless 上の Apache Spark では、データカタログビューのクエリはサポートされません。
- EMR Serverless 上の Apache Hive では、Lake Formation アクセス許可によるテーブルのクエリはサポートされません。
- AWS Glue 5.0 以降では、S3 でバックアップされたデータカタログ内の Iceberg テーブルと Hive テーブルに対するきめ細かなアクセスコントロールがサポートされています。この機能を使用する

と、Apache Spark ジョブの 内の読み取りクエリ AWS Glue のテーブル、行、列、セルレベルのアクセスコントロールを設定できます。

詳細については、「AWS Glue のバージョン」を参照してください。

AWS トランザクションテーブル形式の サービスとサポートされているアクセス許可タイプ

AWS サービス	Iceberg	Hudi	Delta Lake (ネイティブ)	Delta Lake (シンボリックリンクテーブル)
Athena SQL	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションにはフルテーブルアクセスが必要です。	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りおよび作成オペレーションがサポートされます。書き込みオペレーションはサポートされません。	Athena (エンジンバージョン 3) では、テーブル、列、行、セルレベルのアクセス許可によるネイティブ Delta Lake テーブルの読み取りがサポートされます。書き込みオペレーションはサポートされません。	Athena (エンジンバージョン 3) では、テーブル、列、行、セルレベルのアクセス許可によるシンボリックリンク Delta Lake テーブルの読み取りがサポートされます。書き込みオペレーションはサポートされません。
プロビジョニングされたクラスター上の Redshift Spectrum	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションはサポートされません。	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションはサポートされません。	サポートされません。	テーブル、列、行、セルレベルのアクセス許可による、シンボリックリンクマニフェストを介した Lake テーブルの読み取りがサポートされます。書き込みオペレーション

AWS サービス	Iceberg	Hudi	Delta Lake (ネイティブ)	Delta Lake (シンボリックリンクテーブル)
				はサポートされません。
Amazon EMR (EC2) 上の Apache Spark	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションにはフルテーブルアクセスが必要です。	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションにはフルテーブルアクセスが必要です。	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションはサポートされません。	テーブル、列、行、セルレベルのアクセス許可によるテーブルの読み取りがサポートされます。書き込みオペレーションにはフルテーブルアクセスが必要です。
AWS Glue ETL	AWS Glue 5.0 以降では、テーブル、列、行、セルレベルのアクセス許可を持つテーブルの読み取りがサポートされています。	テーブルレベルのアクセス許可によるテーブルの読み取り/書き込みがサポートされます。	テーブルレベルのアクセス許可によるテーブルの読み取り/書き込みがサポートされます。	テーブルレベルのアクセス許可によるテーブルの読み取り/書き込みがサポートされます。

トピック

- Amazon Athena AWS Lake Formation での の使用
- Amazon Redshift Spectrum AWS Lake Formation での の使用
- AWS Lake Formation での の使用 AWS Glue
- Amazon EMR AWS Lake Formation での の使用
- Quick AWS Lake Formation での の使用
- AWS CloudTrail Lake AWS Lake Formation での の使用

Amazon Athena AWS Lake Formation での の使用

Amazon Athena は、Amazon S3 に保存された構造化データ、半構造化データ、および非構造化データの分析に役立つサーバーレスのクエリサービスです。Athena SQL を使用して、CSV、JSON、Parquet、Avro データ形式のデータをクエリできます。Athena SQL は、Apache Hive、Apache Hudi、Apache Iceberg などのテーブル形式もサポートしています。Athena は、Amazon S3 のデータセットのメタデータストアを保存するために、 AWS Glue Data Catalog と統合します。Athena は Lake Formation を使用して、これらのデータセットのアクセスコントロールポリシーを定義および管理できます。

ここでは、Athena で Lake Formation を使用できるいくつかの一般的なユースケースを示します。

- Athena から Data Catalog リソース (データベースとテーブル) にアクセスするための Lake Formation のアクセス許可を使用します。名前付きリソース方式または LF タグのいずれかを使用して、データベースとテーブルに対するアクセス許可を定義できます。詳細については、以下を参照してください。
  - 名前付きリソース方式を使用したデータベースのアクセス許可の付与
  - Lake Formation のタグベースのアクセス制御

> **Note**
>
> Lake Formation アクセス許可は、Athena SQL を使用して Amazon S3 のソースデータとデータカタログ内のメタデータをクエリする場合にのみ適用されます。
> Athena Spark では、Lake Formation アクセス許可によるデータカタログテーブルのクエリはサポートされません。Lake Formation のアクセス許可は、データベースとテーブルに対する読み取りオペレーションおよび書き込みオペレーションの両方をサポートします。

> **Note**
>
> LF タグを使用して Data Catalog リソースに対するアクセス許可を管理する場合、データフィルターを適用することはできません。

- Lake Formation でのデータフィルター を使用し、列、行、およびセルレベルでアクセス許可を付与して Amazon S3 データレイクのテーブルを保護することで、クエリ結果を制御できます。Amazon Athena ユーザーガイドのパーティション射影の制限に関する項目を参照してください。

- フェデレーションクエリを実行する際に、SAML ベースの Athena ユーザーが利用できるデータに細粒度のアクセスコントロールを適用します。

  Athena JDBC および ODBC ドライバーは、SAML ベースの ID プロバイダー (IdP) を使用したデータソースへのフェデレーションアクセスの設定をサポートします。Lake Formation と統合されたクイックを既存の IAM ロールまたは SAML ユーザーまたはグループとともに使用して、Athena クエリ結果を視覚化します。

  > **Note**
  >
  > SAML ユーザーおよびグループに対する Lake Formation のアクセス許可は、JDBC または ODBC ドライバーを使用して Athena にクエリを送信する場合のみ適用されます。

  詳細については、「Athena へのフェデレーションアクセスのための Lake Formation と Athena JDBC および ODBC ドライバーの使用」を参照してください。

  > **Note**
  >
  > 現在、以下のリージョンでは Lake Formation での SAML アイデンティティへのアクセスの認可はサポートされていません。
  >
  > - 中東 (バーレーン) – me-south-1
  > - アジアパシフィック (香港) – ap–east–1
  > - アフリカ (ケープタウン) – af-south-1
  > - 中国 (寧夏) – cn-northwest-1
  > - アジアパシフィック (大阪) – ap-northeast-3

- 別のアカウントのテーブルをクエリする場合は、Lake Formation でのクロスアカウントデータ共有 を使用します。

> **Note**
>
> `Views` に対して Lake Formation アクセス許可を使用する際の制限の詳細については、「考慮事項と制限事項」を参照してください。

トランザクションテーブル形式のサポート

Lake Formation アクセス許可を適用すると、Amazon S3 ベースのデータレイク内のトランザクションデータを保護できます。以下の表は、Athena と Lake Formation のアクセス許可でサポートされているトランザクションテーブル形式を示しています。Lake Formation は、Athena ユーザーがクエリを実行したときにこれらのアクセス許可を適用します。

テーブル形式	説明と許可されるオペレーション	Athena でサポートされている Lake Formation のアクセス許可
Apache Hudi	増分データ処理とデータパイプラインの開発を簡素化するために使用される形式。 Athena は、Copy on Write (CoW) とMerge On Read (MoR) の両方の Hudi テーブルタイプについて、Amazon S3 データセットの Apache Hudi テーブル形式を使用した作成および読み取りオペレーションをサポートしていますが、Hudi テーブルへの書き込みオペレーションはサポートしていません。 Athena を使用して Hudi データセットへのクエリを実行します。	「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」に従って、テーブル、列、行、セルレベルのアクセス許可を使用して Hudi テーブルを保護します。
Apache Iceberg	大量のファイルのコレクションをテーブルとして管理し、レコードレベルの挿入、更新、削除、タイムトラベルクエリなどの最新の分析データレイクオペレーションをサ	テーブル、列、行、セルレベルのアクセス許可がサポートされています。現在、Lake Formation は、オープンテーブルフォーマットのテーブルに対する VACUUM、MERGE、UPDATE、OPTIMIZ

テーブル形式	説明と許可されるオペレーション	Athena でサポートされている Lake Formation のアクセス許可
	ポートするオープンテーブル形式。 Athena による Iceberg テーブルのサポートの詳細については、「Iceberg テーブルの使用」を参照してください。	などの書き込み操作の権限管理をサポートしていません。

テーブル形式	説明と許可されるオペレーション	Athena でサポートされている Lake Formation のアクセス許可
Linux Foundation Delta Lake	Delta Lake は、一般的に Amazon S3 または File system distribuito Hadoop (HDFS) 上に構築される最新のデータレイクアーキテクチャの実装を支援するオープンソースプロジェクトです。 Athena は、Delta Lake テーブル AWS Glue Data Catalog から でシンボリックリンクベースのマニフェストテーブル定義を使用して作成された Delta Lake テーブルをサポートします。 詳細については、「クローラーを使用して Delta Lake テーブル AWS Glue をクロールする」を参照してください。 Athena (エンジンバージョン 3) は、ネイティブの Delta Lake テーブルの読み取りをサポートしています。 詳細については、「クローラによるネイティブ Delta Lake AWS Glue テーブルサポートの紹介」を参照してください。	シンボリックリンクテーブルとネイティブ Delta Lake テーブルでは、テーブル、列、行、およびセルレベルのアクセス許可がサポートされています。

その他のリソース

ブログ投稿、ビデオ、ワークショップ

- Amazon Athena を使用した、Amazon S3 データレイクの Apache Hudi データセットへのクエリの実行
- Amazon Athena、および を使用して Apache Iceberg データレイクを構築する AWS Glue
- Athena と Apache Iceberg を使用した、Amazon S3 での挿入、更新、削除
- LF-Tag ベースのアクセスコントロール データレイクのクエリに関する Lake Formation ワークショップ。

Amazon Redshift Spectrum AWS Lake Formation での の使用

Amazon Redshift Spectrum では、Amazon Redshift クラスターノードにデータをロードすることなく、Amazon S3 データレイクのデータのクエリと取得を実行できます。

Redshift Spectrum は、Lake Formation で有効になっている外部 AWS Glue データカタログを登録する 2 つの方法をサポートしています。

- データカタログへのアクセス許可を持つ、クラスターにアタッチされた IAM ロールの使用

  IAM ロールを作成するには、以下の手順で説明されているステップに従います。

  へのアクセスの制御 AWS Glue Data Catalog
- 外部 AWS Glue Data Catalog リソースへのアクセスを管理するように設定されたフェデレーション IAM ID。

  Redshift Spectrum は、フェデレーション IAM ID を使用したLake Formation テーブルのクエリをサポートしています。IAM ID は、IAM ユーザーまたは IAM ロールとすることができます。Redshift Spectrum での IAM ID フェデレーションの詳細については、「フェデレーション ID を使用して、ローカルリソースと Amazon Redshift Spectrum の外部テーブルへの Amazon Redshift アクセスを管理する」を参照してください。

Lake Formation と Redshift Spectrum の統合により、データを Lake Formation に登録した後に、テーブルに対して行、列、およびセルレベルのアクセスコントロールのアクセス許可を定義できます。

詳細については、「 で Redshift Spectrum を使用する AWS Lake Formation」を参照してください。

Redshift Spectrum は、Lake Formation が管理する外部スキーマテーブルの読み取りまたは SELECT クエリをサポートしています。

詳細については、「Redshift Spectrum 用の外部スキーマの作成」を参照してください。

トランザクションテーブルタイプのサポート

この表は、Redshift Spectrum でサポートされているトランザクションテーブル形式と、該当する Lake Formation アクセス許可を示しています。

サポートされるテーブル形式

テーブル形式	説明と許可されるオペレーション	Redshift Spectrum でサポートされる Lake Formation のアクセス許可
Apache Hudi	増分データ処理とデータパイプラインの開発を簡素化するために使用される形式。 Redshift Spectrum は、Amazon S3 の Apache Hudi Copy on Write (CoW) テーブル形式を使用した挿入、削除、アップサートの書き込みオペレーションをサポートしています。 詳細については、「Apache Hudi で管理されるデータの外部テーブルの作成」を参照してください。	「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」に従って、テーブル、列、行、セルレベルのアクセス許可を使用して Hudi テーブルを保護します。
Apache Iceberg	オープンテーブル形式は、大量のファイルのコレクションをテーブルとして管理し、レコードレベルの挿入、更新、削除、タイムトラベルクエリなどの最新の分析データレイ	Redshift Spectrum は、Apache Iceberg テーブルのクエリをサポートしています。

テーブル形式	説明と許可されるオペレーション	Redshift Spectrum でサポートされる Lake Formation のアクセス許可
	クオペレーションをサポートします。 詳細については、「Amazon Redshift での Apache Iceberg テーブルの使用」を参照してください。	
Linux Foundation Delta Lake	Delta Lake は、一般的に Amazon S3 または File system distribuito Hadoop (HDFS) 上に構築される最新のデータレイクアーキテクチャの実装を支援するオープンソースプロジェクトです。 Redshift Spectrum は、Delta Lake テーブルのクエリをサポートしています。詳細については、「Delta Lake で管理される外部テーブルの作成」を参照してください。	テーブル、列、行、セルレベルのアクセス許可がサポートされています。

その他のリソース

ブログ投稿およびワークショップ

- Amazon Redshift Spectrum で最新のデータアーキテクチャを有効に AWS Lake Formation しながら、 を使用してデータレイクのガバナンスを一元化する
- Redshift Spectrum を使用して、Amazon S3 データレイクで Apache HUDI Copy On Write 時にコピー (CoW) テーブルをクエリする

AWS Lake Formation での の使用 AWS Glue

データエンジニアと DevOps プロフェッショナルは、Apache Spark AWS Glue で抽出、変換、ロード (ETL) を使用して Amazon S3 のデータセットで変換を実行し、変換されたデータを分析、機械学習、アプリケーション開発用のデータレイクとデータウェアハウスにロードします。複数のチームが Amazon S3 の同じデータセットにアクセスする場合、それぞれのロールに基づいてアクセス許可を付与および制限することが不可欠です。

AWS Lake Formation は 上に構築されており AWS Glue、サービスは次の方法でやり取りします。

- Lake Formation と AWS Glue は同じ Data Catalog を共有しています。
- 以下の Lake Formation コンソール機能は、AWS Glue コンソールを呼び出します。
  - ジョブ – 詳細については、AWS Glue デベロッパーガイドの「ジョブを追加する」を参照してください。
  - クローラー – 詳細については、AWS Glue デベロッパーガイドの「クローラーを使用したテーブルのカタログ化」を参照してください。
- Lake Formation のブループリントを使用するときに生成されるワークフローは、AWS Glue ワークフローです。これらのワークフローは、Lake Formation コンソールと AWS Glue コンソールの両方で表示および管理できます。
- Lake Formation では機械学習変換が提供されており、これらは AWS Glue API 操作上に構築されています。機械学習変換は AWS Glue コンソールで作成し、管理します。詳細については、「AWS Glue デベロッパーガイド」の「機械学習変換」を参照してください。

Lake Formation の細粒度のアクセスコントロールを使用して、既存のデータカタログリソースと Amazon S3 データロケーションを管理できます。

> **Note**
>
> AWS Glue 5.0 以降では、S3 でサポートされている Iceberg テーブルと Hive テーブルに対するきめ細かなアクセスコントロールがサポートされています。この機能を使用すると、Apache Spark ジョブの 内の読み取りクエリ AWS Glue のテーブル、行、列、セルレベルのアクセスコントロールを設定できます。

トランザクションテーブルタイプのサポート

Lake Formation アクセス許可を適用すると、Amazon S3 ベースのデータレイク内のトランザクションデータを保護できます。次の表に、 でサポートされているトランザクションテーブル形式 AWS Glue と Lake Formation アクセス許可を示します。Lake Formation は AWS Glue 、オペレーションにこれらのアクセス許可を適用します。

サポートされるテーブル形式

テーブル形式	説明と許可されるオペレーション	でサポートされている Lake Formation アクセス許可 AWS Glue
Apache Hudi	増分データ処理とデータパイプラインの開発を簡素化するために使用されるオープンテーブル形式。 例については、「Using the Hudi framework in AWS Glue」を参照してください。	テーブルレベルのアクセス許可は、Hudi テーブルで利用できます。 詳細については、「制限事項」を参照してください。
Apache Iceberg	大量のファイルのコレクションをテーブルとして管理するオープンテーブル形式。 例については、「 での Iceberg フレームワークの使用 AWS Glue」を参照してください。	AWS Glue バージョン 5.0 以降では、Iceberg テーブルの Apache Spark ジョブ AWS Glue の 内の読み取りクエリのテーブル、行、列、セルレベルのアクセスコントロールを設定できます。 詳細については、「制限事項」を参照してください。
Linux Foundation Delta Lake	Delta Lake は、一般的に Amazon S3 または File system distribuito Hadoop (HDFS) 上に構築される最新のデータレイクアーキテクチャ	テーブルレベルのアクセス許可は、Delta Lake テーブルで利用できます。 詳細については、「制限事項」を参照してください。

テーブル形式	説明と許可されるオペレーション	でサポートされている Lake Formation アクセス許可 AWS Glue
	の実装を支援するオープンソースプロジェクトです。 例については、「 での Delta Lake フレームワークの使用 AWS Glue」を参照してください。	

その他のリソース

ブログ投稿とリポジトリ

- AWS Glue コネクタを使用して ACID トランザクションで Apache Iceberg テーブルを読み書きし、タイムトラベルを実行する
- AWS Glue カスタムコネクタを使用した Apache Hudi テーブルへの書き込み
- AWS Cloudformation テンプレートと pyspark コードサンプルの リポジトリ。 AWS Glue、Apache Hudi、Amazon S3 を使用してストリーミングデータを分析します。

Amazon EMR AWS Lake Formation での の使用

Amazon EMR は、Hadoop Map-Reduce、Spark、Hive、Presto などのサポートされているビッグデータフレームワークで任意のカスタムコードを実行することができる柔軟な AWS マネージドクラスタープラットフォームです。また、Organizations は Amazon EMR を使用して、高度に分散されたクラスター全体でバッチとストリームの両方のデータ処理アプリケーションを実行します。Amazon EMR 上の Apache Spark を使用すると、Lake Formation によってアクセス許可が管理されているデータベースやテーブルでデータ変換とカスタムコードを実行できます。

Amazon EMR をデプロイするには、3 つのオプションがあります。

- EMR on EC2
- EMR Serverless
- Amazon EMR on EKS

詳細については、「Amazon EMR と Lake Formation の統合」または「EMR Serverless と を使用したきめ細かなアクセスコントロール AWS Lake Formation」を参照してください。

トランザクションテーブル形式のサポート

Amazon EMR リリース 6.15.0 以降では、Spark SQL を使用してデータを読み書きする際の Apache Hudi、Apache Iceberg、および Delta Lake のテーブル形式に対する Lake Formation のテーブル、行、列、およびセルレベルのアクセスコントロール許可がサポートされています。

制限については、「Amazon EMR での Lake Formation の使用に関する考慮事項」を参照してください。

サポートされるテーブル形式

テーブル形式	説明と許可されるオペレーション	Amazon EMR でサポートされている Lake Formation 許可
Apache Hudi	増分データ処理とデータパイプラインの開発を簡素化するために使用されるオープンテーブル形式。 サポートされているオペレーションのリストについては、「Apache Hudi と Lake Formation」を参照してください。	Amazon EMR は、Apache Hudi を使用した、テーブル、行、列、セルレベルのアクセスコントロールをサポートしています。
Apache Iceberg	大量のファイルのコレクションをテーブルとして管理するオープンテーブル形式。 サポートされているオペレーションのリストについては、「Apache Iceberg と Lake Formation」を参照してください。	Amazon EMR は、Apache Iceberg を使用した、テーブル、行、列、セルレベルのアクセスコントロールをサポートしています。
Linux Foundation Delta Lake	Delta Lake は、一般的に Amazon S3 または File	Amazon EMR は、Delta Lake テーブルでのテーブル、行、

テーブル形式	説明と許可されるオペレーション	Amazon EMR でサポートされている Lake Formation 許可
	system distribuito Hadoop (HDFS) 上に構築される最新のデータレイクアーキテクチャの実装を支援するオープンソースプロジェクトです。 サポートされているオペレーションのリストについては、「Delta Lake と Lake Formation」を参照してください。	列、セルレベルのアクセスコントロールをサポートしています。

その他のリソース

ユーザーガイド、ブログ投稿、ワークショップ

- Integration with Amazon EMR using Runtime Roles (ランタイムロールを使用した Amazon EMR との統合)
- Get a quick start with Apache Hudi, Apache Iceberg, and Delta Lake with Amazon EMR on EKS (EKS でのAmazon EMR を使って、Apache Hudi、Apache Iceberg、および Delta Lake の使用を迅速に開始)
- EMR Serverless での Delta Lake OSS の使用

Quick AWS Lake Formation での の使用

クイックは、Athena を使用して Amazon S3 の Lake Formation アクセス許可によって管理されるデータセットの探索をサポートします。

Quick の Standard Edition ユーザーと Enterprise Edition ユーザーの両方が Lake Formation と統合されますが、少し異なります。

- Enterprise Edition – 個々の Quick ユーザーとグループに、データベースとテーブルにアクセスするためのきめ細かなアクセスコントロール (FGAC) アクセス許可を付与します。

- Standard エディション – IAM ロールにデータベースとテーブルにアクセスするアクセス許可を付与します。

> **Note**
>
> デフォルトでは、Quick は という名前のロールを使用しますaws-quicksight-service-role-v0。Quick が Athena にアクセスできるようにする必要なアクセス許可を持つカスタムロールを定義することもできます。

詳細については、「 を介した接続の承認 AWS Lake Formation」を参照してください。

その他のリソース

ブログ記事

- でクイック作成者のきめ細かなアクセス許可を有効にする AWS Lake Formation
- AWS Lake Formation と Quick を使用してデータを安全に分析する

AWS CloudTrail Lake AWS Lake Formation での の使用

AWS CloudTrail Lake は、 できめ細かなアクセス許可 Amazon Athena を持つ を使用したイベントデータストアの探索をサポートしています AWS Lake Formation。

> **Note**
>
> CloudTrail Lake はクエリのみを実行できます Amazon Athena。

CloudTrail Lake イベントデータストアを Lake Formation に登録するには、「イベントデータストアのフェデレーション」を参照してください。

を使用した AWS Lake Formation API コールのログ記録 AWS CloudTrail

AWS Lake Formation は AWS CloudTrail、Lake Formation のユーザー、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、すべての Lake Formation API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、Lake Formation コンソールからの呼び出し、 AWS Command Line Interface、Lake Formation API アク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、Lake Formation のイベントを含めた CloudTrail イベントの Amazon S3 バケットへの継続的な配信を有効にすることができます。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 によって収集された情報を使用して、Lake Formation に対して行われたリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエストが行われた日時、および追加の詳細を確認することができます。

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。

CloudTrail 内の Lake Formation 情報

CloudTrail は、新しい AWS アカウントを作成するとデフォルトで有効になります。Lake Formation でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントとして記録されます。イベントは、あらゆるソース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、およびリクエストパラメータに関する情報が含まれています。さらに、すべてのイベントまたはログエントリには、リクエストの生成元に関する情報も含まれています。アイデンティティ情報は、以下を判断するために役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

AWS アカウントの最近のイベントを表示、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。

Lake Formation のイベントなど、 AWS アカウント内のイベントの継続的な記録については、証跡を作成します。追跡を有効にすることで、CloudTrail でログファイルを Amazon S3 バケットに送信できるようになります。デフォルトでは、コンソールで追跡を作成すると、すべての AWS リージョンに追跡が適用されます。証跡は、 AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理 Amazon Athenaするように、 などの他の AWS サービスを設定できます。CloudTrail は、Amazon CloudWatch Logs および CloudWatch Events にログファイルを配信することもできます。

詳細については、以下を参照してください。

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- 「CloudTrail の Amazon SNS 通知の設定」
- 「複数のリージョンから CloudTrail ログファイルを受け取る」および「複数のアカウントから CloudTrail ログファイルを受け取る」

Lake Formation イベントについて

Lake Formation のすべての API アクションは CloudTrail によってログに記録され、 AWS Lake Formation デベロッパーガイドに記載されています。例えば `PutDataLakeSettings`、`GrantPermissions`、および `RevokePermissions` アクションに対する呼び出しは、CloudTrail ログファイルにエントリを生成します。

以下は、`GrantPermissions` アクションに対する CloudTrail イベントを示す例です。このエントリには、許可を付与したユーザー (`datalake_admin`)、許可が付与されたプリンシパル (`datalake_user1`)、および付与された許可 (`CREATE_TABLE`) が含まれています。このエントリは、`resource` 引数にターゲットデータベースが指定されていなかったために、付与が失敗したことも示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAZKE67KM3P775X74U2",
        "arn": "arn:aws:iam::111122223333:user/datalake_admin",
        "accountId": "111122223333",
        "accessKeyId": "...",
```

```
        "userName": "datalake_admin"
    },
    "eventTime": "2021-02-06T00:43:21Z",
    "eventSource": "lakeformation.amazonaws.com",
    "eventName": "GrantPermissions",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.65",
    "userAgent": "aws-cli/1.19.0 Python/3.6.12
 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 botocore/1.20.0",
    "errorCode": "InvalidInputException",
    "errorMessage": "Resource must have one of the have either the catalog, table or
 database field populated.",
    "requestParameters": {
        "principal": {
            "dataLakePrincipalIdentifier": "arn:aws:iam::111122223333:user/
datalake_user1"
        },
        "resource": {},
        "permissions": [
            "CREATE_TABLE"
        ]
    },
    "responseElements": null,
    "requestID": "b85e863f-e75d-4fc0-9ff0-97f943f706e7",
    "eventID": "8d2ccef0-55f3-42d3-9ede-3a6faedaa5c1",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

以下は、GetDataAccess アクションに対する CloudTrail ログエントリを示す例です。プリンシパルは、この API を直接コール呼び出しません。むしろ、プリンシパルまたは統合 AWS サービスGetDataAccessが、Lake Formation に登録されているデータレイクロケーション内のデータにアクセスするための一時的な認証情報をリクエストするたびに、 がログに記録されます。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId": "AROAQGFTBBBGOBWV2EMZA:GlueJobRunnerSession",
        "accountId": "111122223333"
```

```
    },
    "eventSource": "lakeformation.amazonaws.com",
    "eventName": "GetDataAccess",
...
...
    "additionalEventData": {
        "requesterService": "GLUE_JOB",
        "lakeFormationPrincipal": "arn:aws:iam::111122223333:role/ETL-Glue-Role",
        "lakeFormationRoleSessionName": "AWSLF-00-GL-111122223333-G13T0Rmng2"
    },
...
}
```

Lake Formation を使用した CloudTrail のデータアクセスのトレース

Lake Formation は、、Amazon EMR、Amazon S3 のテーブルデータロケーションにアクセスするためのクエリエンジン AWS Glueに対して、スコープダウンされた一時的な認証情報を発行します。Amazon S3 Lake Formation は、データロケーションを Lake Formation に登録するときに指定した IAM ロールを使用して、これらの認証情報を提供します。

CloudTrail のログ記録動作の変更 (2026 年 2 月)

2026 年 2 月以前は、クエリエンジンがクエリデータを取得するための`s3:GetObject`呼び出しを行ったときに、CloudTrail ログに Lake Formation 登録 IAM ロールがプリンシパルとして表示されていました。

2026 年 2 月以降、 `s3:GetObject`呼び出しの CloudTrail ログには、クエリを開始した IAM ロールの IAM 一意の識別子 (以降AROA) も含まれます。この識別子は、クエリ、Amazon EMR ジョブ、または ETL ジョブを開始した IAM AWS Glue ロールに対応します。

クエリ開始から Amazon S3 へのデータアクセスのトレース

CloudTrail の各`s3:GetObject`呼び出しは、`lakeformation:GetDataAccess`呼び出しに対応します。両方のログエントリに存在する IAM ロール識別子を使用して、クエリの開始から Amazon S3 のデータアクセスまで、呼び出し元の IAM ロールを追跡できます。

Amazon S3 データイベント CloudTrail ログでソース ID の情報を有効にするには、以下の手順に従います。CloudTrail で Amazon S3 データイベントを有効にするには、Amazon S3バケットとオブジェクトの CloudTrail イベントログ記録を有効にする」を参照してください。

> **Important**
>
> CloudTrail で Amazon S3 データイベントを有効にすると、追加料金が発生します。

CloudTrail ログでソース ID の追跡を有効にするには

1. を使用して、 `get-data-lake-settings` コマンドを実行します。レスポンスで、`Parameters`フィールドを編集して を追加します`"SET_SOURCE_IDENTITY": "TRUE"`。

   `put-data-lake-settings` コマンドを実行します。

   ```
   "Parameters": {
       "CROSS_ACCOUNT_VERSION": "4",
       "SET_SOURCE_IDENTITY": "TRUE"
   },
   ```

2. Amazon S3 データロケーションを Lake Formation に登録するために使用される IAM ロールの信頼ポリシーで、 を追加します`sts:SetSourceIdentity`。

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

次の`s3:GetObject`ログは、CloudTrail ログの例のソース ID 情報を示しています。

```
"sessionContext": {
```

```
    "sessionIssuer": {
        "type": "Role",
        "principalId": "AROA2EXAMPLEXYZ3AB6CDKLM",
        "arn": "arn:aws:iam::111122223333:role/LFRegistrationRole",
        "accountId": "111122223333",
        "userName": "LFRegistrationRole"
    },
    "sourceIdentity": "V2.111122223333.AROA3EXAMPLEPQR7ST9UVWXY"
},
"invokedBy": "glue.amazonaws.com"
```

以下の資料も参照してください。

- CloudTrail ログでの IAM Identity Center ユーザーコンテキストの追跡については、「」を参照してくださいIAM アイデンティティセンターの統合。

Amazon Redshift フェデレーティッドカタログアクセスのモニタリング

の Amazon Redshift フェデレーティッドカタログは、Lake Formation CloudTrail ログにGetDataAccessイベントを生成 AWS Glue Data Catalog しません。データカタログ内の Amazon Redshift テーブルのデータアクセスを追跡するには、GetTable イベントとBatchGetTableイベントをモニタリングします。これにより、テーブル名、データベース名、カタログ ID 情報など、Redshift フェデレーティッドテーブルのメタデータアクセスパターンがキャプチャされます。

以下の資料も参照してください。

- CloudTrail のクロスアカウントロギング

Lake Formation のベストプラクティス、考慮事項、制限事項

このセクションでは、 AWS Lake Formationのベストプラクティス、考慮事項、制限事項をすばやく見つけます。

AWS アカウントのサービスリソースまたはオペレーションの最大数については、「サービスクォータ」を参照してください。

トピック

- クロスアカウントデータ共有のベストプラクティスと考慮事項
- サービスにリンクされたロールの制限
- クロスリージョンのデータアクセスに関する制限
- データカタログビューの考慮事項と制限
- データフィルタリングの制限事項
- ハイブリッドアクセスモードには次の考慮事項と制限事項が適用されます。
- Amazon Redshift データウェアハウスデータを に取り込む際の制限 AWS Glue Data Catalog
- S3 Tables カタログ統合の制限事項
- Hive メタデータストアのデータ共有に関する考慮事項と制限事項
- Amazon Redshift データ共有の制限事項
- IAM アイデンティティセンター 統合の制限事項
- Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項
- 属性ベースのアクセス制御に関する考慮事項、制限、サポートされているリージョン

クロスアカウントデータ共有のベストプラクティスと考慮事項

Lake Formation のクロスアカウント機能を使用すると、ユーザーは分散データレイクを複数の AWS 組織間で安全に共有したり AWS アカウント、別のアカウントの IAM プリンシパルと直接共有したりして、データカタログのメタデータと基盤となるデータにきめ細かなアクセスを提供したりできます。

Lake Formation のクロスアカウントデータ共有を使用するときは、以下のベストプラクティスを検討してください。

- 自分の AWS アカウントのプリンシパルに対して実行できる Lake Formation アクセス許可の付与の数に制限はありません。ただし、Lake Formation は、アカウントが名前付きリソースメソッドで実行できるクロスアカウント許可に AWS Resource Access Manager (AWS RAM) 容量を使用します。 AWS RAM 容量を最大化するには、名前付きリソースメソッドの以下のベストプラクティスに従います。
  - 新しいクロスアカウント付与モード (クロスアカウントバージョン設定の バージョン 3 以降) を使用して、リソースを外部と共有します AWS アカウント。詳細については、「クロスアカウントデータ共有のバージョン設定の更新」を参照してください。
  - AWS アカウントを組織に整理し、組織または組織単位にアクセス許可を付与します。組織または組織単位への付与は、1 つの付与として計上されます。

    組織または組織単位に を付与すると、グラントの AWS Resource Access Manager (AWS RAM) リソース共有の招待を受け入れる必要もなくなります。詳細については、「共有 Data Catalog テーブルとデータベースへのアクセスと表示」を参照してください。
  - データベース内にある多数のテーブルそれぞれに対する許可を付与する代わりに、特別な [All tables] (すべてのテーブル) ワイルドカードを使用して、データベース内のすべてのテーブルに対する許可を付与します。[All tables] (すべてのテーブル) に対する付与は、単一の付与として計上されます。詳細については、「データカタログリソースに対するアクセス許可の付与」を参照してください。

  > **Note**
  >
  > のリソース共有数の上限をリクエストする方法の詳細については AWS RAM、のAWS「サービスクォータ」を参照してくださいAWS 全般のリファレンス。

- Amazon Athena および Amazon Redshift Spectrum クエリエディタに表示するには、そのデータベースの共有データベースへのリソースリンクを作成する必要があります。同様に、Athena と Redshift Spectrum を使用して共有テーブルをクエリできるようにするには、そのテーブルへのリソースリンクを作成する必要があります。そうすることで、リソースリンクがクエリエディタのテーブルリストに表示されます。

  クエリのために多数のテーブルそれぞれに対するリソースリンクを作成する代わりに、[All tables] (すべてのテーブル) ワイルドカードを使用して、データベース内のすべてのテーブルに対する許可を付与することができます。そうすることで、そのデータベースのリソースリンクを作成し、クエリエディタでそのデータベースリソースリンクを選択するときに、クエリのために、そのデータベース内のすべてのテーブルにアクセスできるようになります。詳細については、「リソースリンクの作成」を参照してください。

- 別のアカウントのプリンシパルとリソースを直接共有する場合、受信者アカウントの IAM プリンシパルには、Athena と Amazon Redshift Spectrum を使用して共有テーブルをクエリするためのリソースリンクを作成するアクセス許可がないことがあります。データレイク管理者は、共有されているテーブルごとにリソースリンクを作成する代わりに、プレースホルダーデータベースを作成して `ALLIAMPrincipal` グループに `CREATE_TABLE` アクセス許可を付与できます。その後、受信者アカウントのすべての IAM プリンシパルがプレースホルダーデータベースにリソースリンクを作成し、共有テーブルのクエリを開始できます。

  名前付きリソース方式を使用したデータベースのアクセス許可の付与 でアクセス許可を `ALLIAMPrincipals` に付与する方法については、CLI コマンドの例を参照してください。
- クロスアカウントアクセス許可がプリンシパルに直接付与されると、付与の受信者のみがこれらのアクセス許可を表示できます。受信者の AWS アカウントのデータレイク管理者は、これらの直接許可を表示できません。
- Athena と Redshift Spectrum は列レベルのアクセスコントロールをサポートしますが、これは包含のみで、除外にはサポート適用されません。AWS Glue ETLジョブでは、列レベルのアクセスコントロールはサポートされません。
- リソースが AWS アカウントと共有されている場合、リソースに対するアクセス許可はアカウントのユーザーにのみ付与できます。リソースに対するアクセス許可を他の AWS アカウント、組織 (自分の組織でもない) 、または `IAMAllowedPrincipals`グループに付与することはできません。
- データベースに対する `DROP` または `Super` を外部アカウントに付与することはできません。
- データベースまたはテーブルを削除する前に、クロスアカウント許可を取り消します。それ以外の場合は、 で孤立したリソース共有を削除する必要があります AWS Resource Access Manager。

関連情報

- Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項
- クロスアカウントアクセスに関する追加のルールと制限については、「Lake Formation 許可のリファレンス」の「`CREATE_TABLE`」を参照してください。

サービスにリンクされたロールの制限

サービスにリンクされたロールは、直接リンクされた特殊なタイプの IAM ロールです AWS Lake Formation。このロールには、Lake Formation が AWS サービス間でユーザーに代わってアクションを実行できるようにする事前定義されたアクセス許可があります。

サービスにリンクされたロール (SLR) を使用してデータロケーションを Lake Formation に登録する場合、次の制限が適用されます。

- 一度作成したサービスにリンクされたロールポリシーは変更できません。
- サービスにリンクされたロールは、アカウント間での暗号化されたカタログリソースの共有をサポートしていません。暗号化されたリソースには、特定の AWS KMS キーアクセス許可が必要です。サービスにリンクされたロールには、アカウント間で暗号化されたカタログリソースを操作する機能を含まない事前定義されたアクセス許可があります。
- 複数の Amazon S3 ロケーションを登録する場合、サービスにリンクされたロールを使用すると、IAM ポリシーの制限をすばやく超える可能性があります。これは、サービスにリンクされたロールでは、 によってポリシーが AWS 記述され、すべての登録を含む 1 つの大きなブロックとして増分されるためです。カスタマー管理ポリシーをより効率的に記述したり、複数のポリシーに権限を分散したり、異なるリージョンに異なるロールを使用したりできます。
- Amazon EMR on EC2 は、サービスにリンクされたロールでデータロケーションを登録したデータにアクセスできません。
- サービスにリンクされたロールオペレーションは、 AWS サービスコントロールポリシーをバイパスします。
- データロケーションをサービスにリンクされたロールに登録すると、IAM ポリシーが結果整合性で更新されます。詳細については、「IAM ユーザーガイド」の「IAM ドキュメントのトラブルシューティング」を参照してください。
- サービスにリンクされたロールを使用する場合、および IAM Identity Center を使用している場合、Lake Formation データレイク設定`SET_CONTEXT = TRUE`で を設定することはできません。その理由は、サービスにリンクされたロールには、IAM Identity Center プリンシパルによる`SetContext`監査に必要な信頼できる ID 伝達と互換性のないイミュータブルな信頼ポリシーがあるからです。

クロスリージョンのデータアクセスに関する制限

Lake Formation では、 AWS リージョンをまたいでデータカタログのテーブルにクエリを実行できます。ソースデータベースとテーブルを指す他のリージョンにリソースリンクを作成することで

Amazon Athena、、、Amazon EMR、および AWS Glue ETL を使用して、他のリージョンからリージョンのデータにアクセスできます。クロスリージョンのテーブルアクセスでは、基になるデータやメタデータをデータカタログ内にコピーしなくても、複数のリージョンをまたいでデータにアクセスできます。

クロスリージョンのテーブルアクセスには以下の制限が適用されます。

- Lake Formation では、Amazon Redshift Spectrum を使用して別のリージョンのデータカタログのテーブルにクエリを実行することはサポートしていません。
- Lake Formationコンソールでは、データベースビューとテーブルビューにソースリージョンのデータベース名やテーブル名は表示されません。
- 別のリージョンにある共有データベース内のテーブルを一覧表示するには、まず共有データベースへのリソースリンクを作成し、次にそのリソースリンクを選択して、[テーブルを表示] を選択する必要があります。
- Lake Formation は、SAML ユーザーによるクロスリージョンのリソースリンク呼び出しをサポートしません。
- Lake Formation のクロスリージョン機能には、データ転送に対する追加料金はかかりません。

データカタログビューの考慮事項と制限

データカタログビューに適用される考慮事項と制限事項は、以下のとおりです。

- Lake Formation コンソールからデータカタログビューを作成することはできません。ビューは、AWS CLI または SDK を使用して作成できます。
- 10 個のテーブルからデータカタログビューを作成できます。これはハード制限です。ビューの基盤となる参照テーブルは、同じデータベースまたは同じ AWS アカウント内の異なるデータベースに属すことができます。
- Redshift を使用したデータカタログビューの作成に固有のその他の考慮事項と制限については、「Amazon Redshift データベースデベロッパーガイド」の「データカタログビューの考慮事項と制限」セクションを参照してください。Athena については、Amazon Athenaユーザーガイド」の「データカタログビューの考慮事項と制限」セクションを参照してください。
- データカタログビューは、Lake Formation に登録されているテーブルに対して、ハイブリッドアクセスモードと Lake Formation モードのどちらでも作成できます。

  Lake Formation ハイブリッドアクセスモードでデータカタログビューを使用する場合は、ビューを消費するプリンシパルにアクセスを付与するのではなく、ビューで参照されるベーステーブル

の Lake Formation アクセス許可にプリンシパルをオプトインすることをお勧めします。これにより、 AWS Glue IAM アクセス許可を通じてベーステーブルがコンシューマーに公開されることはありません。

- ビューを共有するクロスアカウント共有バージョンに制限はありません。
- 既に作成されているビューのダイアレクトに `ALTER VIEW` ステートメントを使用すると、データカタログテーブルと同様にビューがバージョニングされます。ビューバージョンは基盤データの変更に伴って変更されるため、以前のビューにロールバックすることはできません。ビューバージョンは削除でき、その場合はデフォルトで次に利用可能な最新バージョンになります。ビューバージョンを変更するときは、選択したビューバージョンのスキーマとデータが同期していることを確認してください。
- 新しいデータカタログ API は導入されません。既存の `CreateTable`、`UpdateTable`、`DeleteTable`、`GetTable` API が更新されます。
- Amazon Redshift は常に、文字列を含むテーブルから varchar 列を含むビューを作成します。他のエンジンからダイアレクトを追加する場合は、文字列の列を明示的な長さで varchar にキャストする必要があります。
- データベース内の `All tables` にデータレイクのアクセス許可を付与すると、被付与者はデータベース内のすべてのテーブルとビューに対するアクセス許可を持つことになります。
- 以下の場合、ビューを作成することはできません。
  - 他のビューを参照する場合。
  - 参照テーブルがリソースリンクの場合。
  - リファレンステーブルが別のアカウントで所有されている場合。
  - 外部の Hive メタストアからの場合。
- クロスアカウント定義ロールは、Redshift Spectrum Dialect ビューではサポートされていません。
- Athena クエリエディタの Athena ダイアレクトのリソースリンクはサポートされていません。Athena ダイアレクトにクロスアカウント定義ロールを使用するには、ベーステーブルを Athena のデータソースとしてホストするアカウントを追加します。

データフィルタリングの制限事項

Data Catalog テーブルに対する Lake Formation 許可を付与するときは、クエリ結果、および Lake Formation と統合されたエンジン内の特定のデータへのアクセスを制限するためのデータフィルタリング仕様を含めることができます。Lake Formation は、列レベルのセキュリティ、行レベルのセキュリティ、およびセルレベルのセキュリティを実現するために、データフィルタリングを使用しま

す。ソースデータにネストされた構造が含まれている場合は、ネストされた列にデータフィルターを定義して適用できます。

列レベルのフィルタリングに関する注意点と制限

列フィルタリングを指定する方法は 3 つあります。

- データフィルターの使用。
- シンプルな列フィルタリングまたはネストされた列フィルタリングの使用。
- タグの使用。

シンプルな列フィルタリングは、包含または除外する列のリストを指定するだけです。Lake Formation コンソール、 API、 の両方が、単純な列フィルタリング AWS CLI をサポートしています。例については、「Grant with Simple Column Filtering」を参照してください。

以下の注意点と制限が列フィルタリングに適用されます。

- AWS Glue 5.0 以降では、Apache Hive および Apache Iceberg テーブルに対してのみ、Lake Formation を介したきめ細かなアクセスコントロールがサポートされています。
- grant オプションと列フィルタリングを伴う SELECT を付与するには、除外リストではなく、包含リストを使用する必要があります grant オプションを使用しない場合は、包含リストまたは除外リストのどちらでも使用することができます。
- テーブルに対する SELECT を列フィルタリングと共に付与するには、テーブルに対する grant オプション付きの SELECT を、行制限なしで付与されている必要があります。すべての行にアクセスできる必要があります。
- grant オプションと列フィルタリングを伴う SELECT をアカウント内のプリンシパルに付与する場合、そのプリンシパルは、別のプリンシパルへの付与時に、同じ列、または付与列のサブセットに対する列フィルタリングを指定する必要があります。grant オプションと列フィルタリングを伴う SELECT を外部アカウントに付与する場合、外部アカウントのデータレイク管理者は、そのアカウント内の別のプリンシパルに、すべての列に対する SELECT を付与することができます。ただし、すべての列に対する SELECT があるとしても、そのプリンシパルに表示されるのは外部アカウントに付与された列のみになります。
- パーティションキーに列フィルタリングを適用することはできません。
- テーブル内の列のサブセットに対する SELECT 許可を持つプリンシパルに、そのテーブルに対する ALTER、DROP、DELETE または INSERT 許可を付与することはできません。テーブルに対する

ALTER、DROP、DELETE または INSERT 許可を持つプリンシパルについては、列フィルタリングを伴う SELECT 許可を付与しても、効果はありません。

以下の注意点と制限が、ネストされた列フィルタリングに適用されます。

- データフィルターでは 5 レベルのネストされたフィールドを含めたり除外したりできます。

  Example

  Col1.Col1_1.Col1_1_1.Col1_1_1_1.Col1_1_1_1_1
- パーティション列内のネストされたフィールドに列フィルタリングを適用することはできません。
- テーブルスキーマに、データフィルター内のネストされたフィールド表現と同じパターンを持つ最上位の列名 ("customer"."address") が含まれている場合 (最上位の列名 `customer` とネストされたフィールド名 `address` を持つネストされた列は、データフィルターで `"customer"."address"` として指定されます)、最上位の列とネストされたフィールドは両方とも包含/除外リストの同じパターンを使用するため、最上位の列またはネストされたフィールドへのアクセスを明示的に指定することはできません。これはあいまいであり、最上位の列を指定しているのか、ネストされたフィールドを指定しているのか、Lake Formation は解決できません。
- 最上位の列またはネストされたフィールドの名前に 1 つの二重引用符が含まれている場合、データセルフィルターの包含リストと除外リスト内のネストされたフィールドへのアクセスを指定するときに、2 つ目の二重引用符を含める必要があります。

  Example

  二重引用符を使用したネストされた列名の例 — `a.b.double"quote`

  Example

  データフィルター内のネストされた列表現の例 — `"a"."b"."double""quote"`

セルレベルのフィルタリングの制限

行レベルおよびセルレベルのフィルタリングに関しては、以下の注意点と制限に留意してください。

- セルレベルのセキュリティは、ネストされた列、ビュー、リソースリンクではサポートされません。

- 最上位の列でサポートされているすべての式は、ネストされた列でもサポートされます。ただし、ネストされた行レベルの式を定義するときは、パーティション列の下のネストされたフィールドを参照しないでください。
- Athena エンジンバージョン 3 または Amazon Redshift Spectrum を使用すると、すべてのリージョンでセルレベルのセキュリティを利用できます。他のサービスでは、セルレベルのセキュリティは、サポートされるリージョン に記載されているリージョンでのみ利用できます。
- `SELECT INTO` ステートメントはサポートされません。
- `array` および `map` データ型は、行フィルター式ではサポートされていません。`struct` データ型はサポートされています。
- テーブルに定義できるデータフィルターの数に上限はありませんが、テーブルあたり 1 つのプリンシパルに対してデータフィルター 100 個という上限があります。
- 行フィルター式があるデータフィルターを適用するには、すべてのテーブル列に対する grant オプション付きの `SELECT` を持っている必要があります。付与が外部アカウントに行われた場合、この制限は外部アカウントの管理者には適用されません。
- プリンシパルがグループのメンバーであり、プリンシパルとグループの両方に行のサブセットに対する許可が付与されている場合、プリンシパルの有効な行の許可は、プリンシパルの許可とグループの許可を合わせたものになります。
- 行レベルおよびセルレベルのフィルタリングでは、テーブルの以下の列名が制限されています。
  - ctid
  - oid
  - xmin
  - cmin
  - xmax
  - cmax
  - tableoid
  - insertxid
  - deletexid
  - importoid
  - redcatuniqueid
- 述語を持つ他のフィルター式と同時に全行フィルター式をテーブルに適用する場合は、全行フィルター式が他のすべてのフィルター式に優先します。

- 行のサブセットに対するアクセス許可が外部 AWS アカウントに付与され、外部アカウントのデータレイク管理者がそのアカウントのプリンシパルにそれらのアクセス許可を付与する場合、プリンシパルの有効なフィルター述語は、アカウントの述語とプリンシパルに直接付与された述語の共通部分です。

  例えば、アカウントに述語 `dept='hr'` を持つ行の許可があり、プリンシパルに `country='us'` の許可を別途付与された場合、プリシパルは `dept='hr'` と `country='us'` の行にのみアクセスすることができます。

セルレベルのフィルタリングの詳細については、「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」を参照してください。

行レベルのセキュリティポリシーで Amazon Redshift Spectrum を使用してテーブルをクエリする際の考慮事項と制限については、「Amazon Redshift データベースデベロッパーガイド」の「RLS ポリシーを使用する際の考慮事項」を参照してください。

ハイブリッドアクセスモードには次の考慮事項と制限事項が適用されます。

ハイブリッドアクセスモードでは、 AWS Glue Data Catalog内のデータベースとテーブルの Lake Formation 許可を柔軟かつ選択的に有効にできます。
ハイブリッドアクセスモードでは、他の既存のユーザーやワークロードのアクセス許可ポリシーを中断することなく、特定のユーザーのセットに Lake Formation 許可を設定できる増分パスが導入されました。

ハイブリッドアクセスモードには次の考慮事項と制限事項が適用されます。

制限事項

- Amazon S3 ロケーション登録の更新 – サービスにリンクされたロールを使用して Lake Formation に登録されているロケーションのパラメータを編集することはできません。
- LF タグを使用する場合のオプトインオプション – LF タグを使用して Lake Formation 許可を付与できる場合は、LF タグがアタッチされているデータベースとテーブルを選択することで、プリンシパルに Lake Formation 許可を連続したステップで適用するようにオプトインできます。
- ハイブリッドアクセスモードでのアクセス – Lake Formation のハイブリッドアクセスモードでのアクセスは、データレイク管理者または読み取り専用管理者のアクセス許可を持つユーザーに限定されます。

- プリンシパルのオプトイン – 現在のところ、プリンシパルをリソースにオプトインできるのはデータレイク管理者ロールだけです。
- データベース内のすべてのテーブルをオプトイン – クロスアカウント付与で、アクセス許可を付与してデータベース内のすべてのテーブルをオプトインする場合、アクセス許可が機能するためにはデータベースもオプトインする必要があります。

考慮事項

- Lake Formation に登録されている Amazon S3 ロケーションをハイブリッドアクセスモードに更新 – Lake Formation に既に登録されている Amazon S3 データロケーションをハイブリッドアクセスモードに変換することは可能ですが、お勧めしません。
- データロケーションがハイブリッドアクセスモードで登録されている場合の API の動作
  - CreateTable — ハイブリッドアクセスモードのフラグとオプトインステータスに関係なく、ロケーションは Lake Formation に登録済みであると見なされます。したがって、ユーザーがテーブルを作成するには、データロケーションへのアクセス許可が必要です。
  - CreatePartition/BatchCreatePartition/UpdatePartition (ハイブリッドに登録されている場所を指すようにパーティションの場所が更新された場合) – Amazon S3 の場所は、ハイブリッドアクセスモードフラグとオプトインステータスに関係なく、Lake Formation に登録されていると見なされます。
  - CreateDatabase/UpdateDatabase (ハイブリッドアクセスモードで登録されたロケーションを指すようにデータベースのロケーションが更新されている場合) – ハイブリッドアクセスモードのフラグとオプトインステータスに関係なく、ロケーションは Lake Formation に登録済みであると見なされます。したがって、ユーザーがデータベースを作成または更新するには、データロケーションへのアクセス許可が必要です。
  - UpdateTable (ハイブリッドアクセスモードで登録されたロケーションを指すようにテーブルのロケーションが更新されている場合) – ハイブリッドアクセスモードのフラグやオプトインステータスに関係なく、ロケーションは Lake Formation に登録済みであると見なされます。したがって、ユーザーがテーブルを更新するには、データロケーションへのアクセス許可が必要です。テーブルロケーションが更新されていないか、Lake Formation に登録されていないロケーションを指している場合、ユーザーはデータロケーションへのアクセス許可を必要とすることなく、テーブルを更新できます。

Amazon Redshift データウェアハウスデータを に取り込む際の制限 AWS Glue Data Catalog

AWS Glue Data Catalogを使用して、Amazon Redshift データウェアハウス内の分析データをカタログ化し、アクセスを管理できます。以下の制限が適用されます。

- フェデレーティッドカタログレベルではクロスアカウント共有はサポートされていません。ただし、フェデレーティッドカタログ内から AWS アカウント間で個々のデータベースとテーブルを共有できます。
- フェデレーティッドカタログ内のデータベースまたはテーブルを 間で共有するには、クロスアカウントバージョン設定バージョン 4 AWS アカウント以降が必要です。
- データカタログでは、最上位レベルのカタログの作成のみがサポートされます。
- カタログの説明は Redshift マネージドストレージ (RMS) でのみ更新できます。
- フェデレーティッドカタログおよびフェデレーティッドカタログ内のデータベースとテーブルに対するアクセス許可を `IAMAllowedPrincipals` グループに設定することはサポートされていません。

- Athena、Amazon EMR Spark などのエンジンからのカタログに対するデータ定義言語 (DDL) オペレーション (カタログ設定の構成を含む) はサポートされていません。
- Athena を使用した RMS テーブルでの DDL オペレーションの実行はサポートされていません。
- マテリアライズドビューの作成は、Athena、Apache Spark、、 AWS Glue Data Catalogまたは Amazon Redshift コンシューマー経由であってもサポートされていません。
- Athena は複数のカタログの利用はサポートしていません。一度に 1 つの特定のカタログにのみ接続できます。Athena は、複数のカタログ間で同時にアクセスまたはクエリを実行することはできません。
- Athena および Amazon Redshift を介した Iceberg テーブルのタグ付けおよび分岐オペレーションはサポートされていません。
- RMS テーブルでの Time Travel はサポートされていません。
- データレイクテーブルを含むマルチレベルカタログはサポートされていません。データレイクテーブルで使用する Amazon S3 に保存されているすべてのデータは、デフォルトで存在する必要があり AWS Glue Data Catalog、複数レベルのカタログに整理することはできません。
- Amazon Redshift では、登録された名前空間にデータ共有は追加されません。クラスターと名前空間は同義です。クラスターを に公開すると AWS Glue Data Catalog、新しいデータを追加できなくなります。

- Amazon EMR on EC2 は、RMS テーブルと Amazon S3 テーブル間の結合をサポートしていません。EMR Serverless のみが、この機能をサポートしています。
- 外部スキーマとテーブルはサポートされていません。
- RMS テーブルには、 AWS Glue Iceberg REST Catalog の拡張エンドポイントからのみアクセスできます。
- Hive テーブルは、 AWS Glue Iceberg REST Catalog に接続されたサードパーティーエンジンからはアクセスできません。
- Spark を介した RMS テーブルの read_committed 分離レベルがサポートされます。
- Redshift データベース名は、 では大文字と小文字が区別されず AWS Glue Data Catalog、128 文字に制限され、ダッシュ (-) とアンダースコア (_) を含む英数字を使用できます。
- カタログ名は大文字と小文字が区別されず、50 文字に制限され、ダッシュ (-) とアンダースコア (_) を含む英数字を使用できます。
- Amazon Redshift では、Lake Formation SQL 形式の GRANT コマンドと REVOKE コマンドを使用して AWS Glue Data Catalogに公開されたテーブルに対するアクセス許可を管理することはサポートされていません。
- プロデューサー (ソース) Amazon Redshift クラスターにアタッチされている行レベルのセキュリティポリシーと動的データマスキングポリシーは適用されません。代わりに、Lake Formation で定義されたアクセス許可が共有データに適用されます。
- テーブルリンクでのデータ定義言語 (DDL) およびデータ操作言語 (DML) オペレーションの実行はサポートされていません。
- 予約キーワードが適切にエスケープされない場合、失敗またはエラーが発生します。
- 複数カタログのシナリオでのデータの暗号化はサポートされていません。

S3 Tables カタログ統合の制限事項

Amazon S3 Tables は AWS Glue Data Catalog (データカタログ) と統合され、カタログを Lake Formation データの場所として登録します。この登録は、Lake Formation コンソールから、またはサービス API を使用して設定できます。

> **Note**
>
> プリンシパルタグに基づいて IAM ユーザーと IAM ロールを制限する IAM または S3 Tables リソースベースのポリシーがある場合は、Lake Formation が S3 データ

(LakeFormationDataAccessRole など) にアクセスするために使用するのと同じプリンシパルタグを IAM ロールにアタッチし、このロールに必要なアクセス許可を付与する必要があります。これは、タグベースのアクセスコントロールポリシーが S3 Tables 分析統合で正しく機能するために必要です。

S3 Tables カタログと Data Catalog および Lake Formation の統合には、次の制限が適用されます。

- AWS Glue および Lake Formation は、大文字と小文字が混在する列名をサポートしておらず、すべての列名を小文字に変換します。そのため、小文字に変換した場合にテーブル列名が一意になることを確認する必要があります。`customerId` ではなく `customer_id` を使用します。大文字と小文字が混在する列名の使用は、プレビューリリースでのみサポートされていました。
- CreateCatalog API は Amazon S3 でテーブルバケットを作成できません。
- SearchTables API は S3 テーブルを検索できません。

Hive メタデータストアのデータ共有に関する考慮事項と制限事項

AWS Glue Data Catalog メタデータフェデレーション (データカタログフェデレーション) を使用すると、データカタログを Amazon S3 データのメタデータを保存する外部メタストアに接続し、 を使用してデータアクセス許可を安全に管理できます AWS Lake Formation。

Hive データベースから作成されたフェデレーションデータベースには、以下の考慮事項と制限事項が適用されます。

考慮事項

- AWS SAM アプリケーションサポート – が AWS SAM デプロイするアプリケーションリソース (Amazon API Gateway および Lambda 関数) の可用性については、お客様の責任となります。ユーザーがクエリを実行するときに、 AWS Glue Data Catalog と Hive メタストア間の接続が機能していることを確認します。
- Hive メタストアのバージョン要件 — Apache Hive バージョン 3 以降でのみフェデレーションデータベースを作成できます。
- マッピングされたデータベースの要件 – Hive の各データベースは、Lake Formation の新しいデータベースにマッピングする必要があります。
- データベースレベルのフェデレーションサポート – Hive メタストアにはデータベースレベルでのみ接続できます。

- フェデレーションデータベースのアクセス許可 – フェデレーションデータベースまたはフェデレーションデータベース内のテーブルに適用されたアクセス許可は、ソーステーブルまたはデータベースが削除された場合でも保持されます。ソースデータベースまたはテーブルを再作成するとき、アクセス許可を再付与する必要はありません。Lake Formation のアクセス許可を持つフェデレーションテーブルをソースで削除しても、Lake Formation のアクセス許可は引き続き表示され、必要に応じて取り消すことができます。

  ユーザーがフェデレーションデータベースを削除すると、対応するアクセス許可はすべて失われます。同じデータベースを同じ名前で再作成しても、Lake Formation のアクセス許可は回復しません。ユーザーは新しいアクセス許可を再度設定する必要があります。
- フェデレーションデータベースの IAMAllowedPrincipal グループのアクセス許可 – `DataLakeSettings` に基づいて、Lake Formation はすべてのデータベースとテーブルに対するアクセス許可を `IAMAllowedPrincipal` という名前の仮想グループに設定する場合があります。は、IAM プリンシパルポリシーとリソースポリシーを通じて Data Catalog リソースにアクセスできるすべての IAM プリンシパル`IAMAllowedPrincipal`を指します AWS Glue 。これらのアクセス許可がデータベースまたはテーブルに存在する場合、すべてのプリンシパルにデータベースまたはテーブルへのアクセス許可が付与されます。

  ただし、Lake Formation では、フェデレーションデータベース内のテーブルに対する `IAMAllowedPrincipal` アクセス許可は許可されていません。フェデレーションデータベースを作成するときは、必ず `CreateTableDefaultPermissions` パラメータを空のリストとして渡してください。

  詳細については、「データレイクのデフォルト設定の変更」を参照してください。
- クエリでのテーブルの結合 – Hive メタストアテーブルをデータカタログのネイティブテーブルと結合してクエリを実行できます。

制限事項

- AWS Glue Data Catalog と Hive メタストア間のメタデータの同期の制限 – Hive メタストア接続を確立したら、フェデレーティッドデータベースを作成して Hive メタストアのメタデータを と同期する必要があります AWS Glue Data Catalog。フェデレーションデータベースのテーブルは、ランタイム時にユーザーがクエリを実行すると同期されます。
- フェデレーションデータベースでの新規テーブル作成の制限 – フェデレーションデータベースでは新しいテーブルを作成できません。
- データのアクセス許可の制限 – Hive メタストアテーブルビューのアクセス許可のサポートはありません。

Amazon Redshift データ共有の制限事項

AWS Lake Formation を使用すると、Amazon Redshift からデータ共有内のデータを安全に管理できます。Amazon Redshift は、 AWS クラウド内のフルマネージド型のペタバイト規模のデータウェアハウスサービスです。Amazon Redshift では、データ共有機能を使用して、 AWS アカウント間でデータを共有できます。Amazon Redshift データ共有の詳細については、「Amazon Redshift でのデータ共有の概要」を参照してください。

Amazon Redshift データ共有から作成されたフェデレーションデータベースには、以下の注意事項と制限事項が適用されます。

- マッピングされたデータベースの要件 – Amazon Redshift の各データ共有は、Lake Formation の新しいデータベースにマッピングする必要があります。これは、データ共有オブジェクト表現がデータカタログデータベースでフラット化されるときに、一意のテーブル名を維持するために必要です。
- フェデレーションデータベースでの新規テーブル作成の制限 – フェデレーションデータベースでは新しいテーブルを作成できません。
- フェデレーションデータベースのアクセス許可 – フェデレーションデータベースまたはフェデレーションデータベース内のテーブルに適用されたアクセス許可は、ソーステーブルまたはデータベースが削除された場合でも保持されます。ソースデータベースまたはテーブルを再作成するとき、アクセス許可を再付与する必要はありません。Lake Formation のアクセス許可を持つフェデレーションテーブルをソースで削除しても、Lake Formation のアクセス許可は引き続き表示され、必要に応じて取り消すことができます。

  ユーザーがフェデレーションデータベースを削除すると、対応するアクセス許可はすべて失われます。同じデータベースを同じ名前で再作成しても、Lake Formation のアクセス許可は回復しません。ユーザーは新しいアクセス許可を再度設定する必要があります。
- フェデレーションデータベースの IAMAllowedPrincipal グループのアクセス許可 – `DataLakeSettings` に基づいて、Lake Formation はすべてのデータベースとテーブルに対するアクセス許可を `IAMAllowedPrincipal` という名前の仮想グループに設定する場合があります。は、IAM プリンシパルポリシーとリソースポリシーを通じて Data Catalog リソースにアクセスできるすべての IAM プリンシパル`IAMAllowedPrincipal`を指します AWS Glue 。これらのアクセス許可がデータベースまたはテーブルに存在する場合、すべてのプリンシパルにデータベースまたはテーブルへのアクセス許可が付与されます。

  ただし、Lake Formation では、フェデレーションデータベース内のテーブルに対する `IAMAllowedPrincipal` アクセス許可は許可されていません。フェデレーションデータベースを

作成するときは、必ず `CreateTableDefaultPermissions` パラメータを空のリストとして渡してください。

詳細については、「データレイクのデフォルト設定の変更」を参照してください。

- データフィルタリング – Lake Formation では、列レベルと行レベルのフィルタリングを使用して、フェデレーションデータベース内のテーブルにアクセス許可を付与できます。ただし、列レベルのフィルタリングと行レベルのフィルタリングを組み合わせて、フェデレーションデータベース内のテーブルへのアクセスをセルレベルの精度で制限することはできません。
- 大文字と小文字の区別識別子 – Lake Formation が管理する Amazon Redshift データ共有オブジェクトでは、テーブル名と列名は小文字でのみサポートされます。Amazon Redshift データ共有のデータベース、テーブル、列が Lake Formation を使用して共有および管理される場合は、大文字と小文字の区別識別子をオンにしないでください。
- クエリのサポート - Lake Formation によって管理される Amazon Redshift データ共有は Amazon Redshift でクエリできます。Athena では、Lake Formation によって管理される Amazon Redshift データ共有のクエリはサポートされていません。

Amazon Redshift でデータ共有を操作する際の制限事項の詳細については、「Amazon Redshift データベース開発者ガイド」の「データ共有に関する制限事項」を参照してください。

IAM アイデンティティセンター 統合の制限事項

を使用すると AWS IAM アイデンティティセンター、ID プロバイダー (IdPs) に接続し、 AWS 分析サービス全体でユーザーとグループのアクセスを一元管理できます。を IAM アイデンティティセンターで有効なアプリケーション AWS Lake Formation として設定でき、データレイク管理者は AWS Glue Data Catalog リソースの承認されたユーザーとグループにきめ細かなアクセス許可を付与できます。

IAM アイデンティティセンターとの Lake Formation の統合には、以下の制限が適用されます。

- Lake Formation では、IAM アイデンティティセンターのユーザーとグループをデータレイク管理者または読み取り専用管理者として割り当てることはできません。

  IAM Identity Center のユーザーとグループは、Data Catalog の暗号化と復号のために がユーザーに代わって引き受け AWS Glue ることができる IAM ロールを使用している場合、暗号化された Data Catalog リソースをクエリできます。 AWS マネージドキーは、信頼できる ID の伝播をサポートしていません。

- IAM ID センターのユーザーとグループは、IAM アイデンティティセンターによって提供された `AWSIAMIdentityCenterAllowListForIdentityContext` ポリシーにリストされている API オペレーションのみを呼び出すことができます。
- Lake Formation は、データカタログリソースへのアクセスのために、外部アカウントの IAM ロールが IAM アイデンティティセンターのユーザーとグループに代わってキャリアロールとして動作することを許可しますが、アクセス許可を付与できるのは、所有アカウント内のデータカタログリソースに対してだけです。外部アカウント内のデータカタログリソースに対するアクセス許可を IAM アイデンティティセンターのユーザーとグループに付与しようとすると、Lake Formation から「Cross-account grants are not supported for the principal」というエラーがスローされます。
- IAM アイデンティティセンターで Lake Formation を使用する場合、アプリケーション割り当て設定はデフォルトで `false` に設定されます。IAM Identity Center API を使用してこの設定を直接変更する場合は、API を使用してすべてのアプリケーション割り当てを手動で管理する必要があります。Lake Formation は、標準ワークフロー外で行われた割り当ての変更を自動的に同期または管理しないため、データレイク環境内のアクセスパターンと認可フローに影響を与える可能性があります。

Lake Formation のタグベースのアクセスコントロールのベストプラクティスと考慮事項

データカタログデータベース、テーブル、および列へのアクセスを制御するための LF タグは、作成、維持、および割り当てを行うことができます。

Lake Formation のタグベースのアクセスコントロールを使用するときは、以下のベストプラクティスを検討してください。

- すべての LF タグは、データカタログリソースに割り当てられたり、プリンシパルに付与される前に、あらかじめ定義しておく必要があります。

  データレイク管理者は、必要な IAM アクセス許可で LF タグ作成者を作成することによって、タグ管理タスクを委任できます。データエンジニアとアナリストは、LF タグの特性と関係を決定します。その後、LF タグ作成者は、Lake Formation で LF タグを作成して管理します。
- 複数の LF タグをデータカタログリソースに割り当てることができます。特定のキーに対する 1 つの値だけを、特定のリソースに割り当てることができます。

例えば、データベース、テーブル、および列には、module=Orders、region=West、および division=Consumer などを割り当てることができます。module=Orders,Customers を割り当てることはできません。

- リソースの作成時に LF タグをリソースに割り当てることはできません。LF タグを追加できるのは、既存のリソースのみです。
- 単一の LF タグだけではなく、LF タグ式をプリンシパルに付与できます。

  LF タグ式は、以下のようになります (擬似コードを使用)。

  ```
  module=sales AND division=(consumer OR commercial)
  ```

  この LF タグ式を付与されたプリンシパルは、module=sales と、division=consumer または division=commercial のいずれかが割り当てられたデータカタログリソース (データベース、テーブル、および列) にのみアクセスできます。module=sales または division=commercial を持つリソースにプリンシパルがアクセスできるようにする場合は、同じ付与に両方を含めないでください。module=sales と division=commercial それぞれに 1 回ずつ、合計で 2 回付与を行います。

  最もシンプルな LF タグ式は、module=sales など、1 つの LF タグだけで構成されます。
- 複数の値を持つ LF タグに対する許可を付与されたプリンシパルは、それらの値のいずれかを持つデータカタログリソースにアクセスできます。例えば、キーが module で値が orders,customers の LF タグがユーザーに付与される場合、そのユーザーは、module=orders または module=customers が割り当てられたリソースにアクセスできます。
- LF-TBAC 方法を使用してデータカタログリソースに対するデータアクセス許可を付与するには、Grant with LF-Tag expressions アクセス許可が必要です。データレイク管理者と LF タグ作成者は、このアクセス許可を暗黙的に受け取ります。Grant with LFTag expressions アクセス許可を持つプリンシパルは、次の方法でリソースに対するデータアクセス許可を付与できます。
  - 名前付きリソースメソッド
  - LF-TBAC 方法。ただし、同じ LF タグ式のみを使用して。

    例えば、データレイク管理者が以下の付与を行うとします (擬似コードを使用)。

```
GRANT (SELECT ON TABLES) ON TAGS module=customers, region=west,south TO user1 WITH
 GRANT OPTION
```

この場合、user1 は LF-TBAC 方法を使用して、ただし完全な LF タグ式 module=customers, region=west,south を使用して、テーブルに対する SELECT を他のプリンシパルに付与できます。

- LF-TBAC 方式と名前付きリソース方式の両方を使用してリソースに対する許可がプリンシパルに付与される場合、そのプリンシパルがリソースに対して持っている許可は、両方の方式によって付与された許可を結合したものになります。
- Lake Formation は、LF-TBAC 方法を使用した複数のアカウントでの LF タグに対する DESCRIBE および ASSOCIATE の付与と、複数のアカウントでのデータカタログに対するアクセス許可の付与をサポートしています。どちらの場合も、プリンシパルは AWS アカウント ID です。

> **Note**
>
> Lake Formation は、LF-TBAC 方式を使用した組織および組織単位へのクロスアカウント付与はサポートします。この機能を使用するには、[Cross account version settings] (クロスアカウントのバージョン設定) を [Version 3] (バージョン 3) に更新する必要があります。

詳細については、「Lake Formation でのクロスアカウントデータ共有」を参照してください。

- 1 つのアカウントで作成されたデータカタログリソースは、同じアカウントで作成された LF タグを使用してのみタグ付けできます。あるアカウントで作成された LF タグを別のアカウントの共有リソースに関連付けることはできません。
- Lake Formation タグベースのアクセスコントロール (LF-TBAC) を使用して Data Catalog リソースへのクロスアカウントアクセスを許可するには、 AWS アカウントの Data Catalog リソースポリシーに追加する必要があります。詳細については、「前提条件」を参照してください。
- LF タグのキーと LF タグの値の長さは 50 文字以下にする必要があります。
- データカタログリソースに割り当てることができる LF タグの最大数は 50 個です。
- 次の制限はソフト制限です。
  - 作成できる LF タグの最大数は 10,000 です。
  - LF タグに定義できる値の最大数は 1,000 個です。
- タグのキーと値はすべて、保存されるときに小文字に変換されます。

- LF タグの 1 つの値だけを、特定のリソースに割り当てることができます。
- 単一の付与で複数の LF タグがプリンシパルに付与される場合、このプリンシパルはすべての LF タグを持つデータカタログリソースのみにアクセスできます。
- LF タグ式の評価結果はテーブル列のサブセットのみへのアクセスであったが、一致があるときに付与される Lake Formation アクセス許可が、列全体へのアクセスを必要とするアクセス許可 (つまり、`Alter`、`Drop`、`Insert` または `Delete`) の 1 つである場合、これらのアクセス許可のいずれも付与されません。その代わり、`Describe` のみが付与されます。付与された許可が `All` (`Super`) である場合は、`Select` と `Describe` のみが付与されます。
- ワイルドカードは LF タグには使用できません。LF タグをテーブルのすべての列に割り当てるには、テーブルに LF タグを割り当てます。これにより、テーブルのすべての列が LF タグを継承します。LF タグをデータベースのすべてのテーブルに割り当てるには、データベースに LF タグを割り当てます。これにより、データベース内のすべてのテーブルがその LF タグを継承します。
- アカウントに最大 1000 個の LF タグ式を作成できます。
- 最大 50 個の LF タグ式を使用して、データカタログリソースのプリンシパルにアクセス許可を付与できます。
- インライン LF タグ式に対するアクセス許可を付与または取り消す場合、LF タグ式のサイズは 900 バイトを超えることはできません。さらに大きな LF タグ式に対するアクセス許可を付与するには、保存された LF タグ式を使用します。詳細については、「LF タグ式の作成」を参照してください。
- フェデレーティッドカタログの LF タグサポートの一般提供リリース前に作成された既存の Redshift フェデレーティッドカタログに LF タグを追加するには、 AWS サポートチームに連絡してサポートを依頼する必要があります。

属性ベースのアクセス制御に関する考慮事項、制限、サポートされているリージョン

属性ベースのアクセス制御 (ABAC) には、以下の考慮事項と制限が適用されます。

- ABAC は、LF タグポリシーを使用したアクセス許可の付与をサポートしていません。
- 付与可能なアクセス許可は ABAC では使用できません。
- ABAC は、IAM アイデンティティセンターユーザーへのアクセス許可の付与をサポートしていません。

- Lake Formation のテーブルで ABAC 許可を使用する場合、Lake Formation は親データベースまたはカタログに DESCRIBE アクセス許可を付与しません。これは、Lake Formation が親リソースに暗黙的な DESCRIBE アクセス許可を提供する非 ABAC シナリオとは異なります。
- AmazonDataZoneProject タグキーを持つすべてのプリンシパルは、常にすべてのデータカタログリソースに対して Lake Formation にオプトインされたものとして扱われます。
- ABAC は文字列属性のみをサポートします。

Lake Formation のトラブルシューティング

AWS Lake Formation の使用中に問題が発生した場合は、このセクションのトピックを参照してください。

トピック

- 一般的なトラブルシューティング
- クロスアカウントアクセスのトラブルシューティング
- ブループリントとワークフローのトラブルシューティング
- の既知の問題 AWS Lake Formation
- エラーメッセージを更新しました

一般的なトラブルシューティング

この情報を使用して、さまざまな Lake Formation 問題の診断と修正に役立ててください。

エラー: Insufficient Lake Formation permissions on <Amazon S3 location> (<Amazon S3 のロケーション> に対する Lake Formation 許可が不十分です)

Data Catalog リソースがポイントする Amazon S3 ロケーションに対するデータロケーション許可がないまま、そのリソースの作成または変更が試行されました。

Data Catalog データベースまたはテーブルが Amazon S3 のロケーションをポイントする場合は、Lake Formation の `CREATE_TABLE` または `ALTER` 許可を付与するときに、その口ケーションに対する `DATA_LOCATION_ACCESS` 許可も付与する必要があります。外部のアカウントまたは組織にこれらの許可を付与している場合は、grant オプションを含める必要があります。

これらの許可が外部アカウントに付与されたら、そのアカウントのデータレイク管理者は、アカウント内のプリンシパル (ユーザーまたはロール) に許可を付与する必要があります。別のアカウントから受け取った`DATA_LOCATION_ACCESS`アクセス許可を付与する場合は、所有者アカウントのカタログ ID (AWS アカウント ID) を指定する必要があります。所有者アカウントは、ロケーションを登録したアカウントです。

詳細については、「基盤となるデータのアクセスコントロール」および「データロケーション許可の付与」を参照してください。

エラー:「Insufficient encryption key permissions for Glue API」(Glue API の暗号化キー許可が不十分です)

暗号化されたデータカタログの AWS KMS 暗号化キーに対する AWS Identity and Access Management (IAM) アクセス許可のない Lake Formation アクセス許可を付与しようとしました。

マニフェストを使用する自分のクエリ Amazon Athena または Amazon Redshift クエリが失敗している

Lake Formation は、マニフェストを使用するクエリをサポートしません。

エラー:「Insufficient Lake Formation permission(s): Required create tag on catalog」(Lake Formation 許可が不十分です: カタログに対する必須の create タグ)

ユーザー/ロールは、データレイク管理者である必要があります。

無効なデータレイク管理者を削除するとエラーが発生します

無効なデータレイク管理者 (データレイク管理者として定義された削除済み IAM ロール) をすべて同時に削除する必要があります。無効なデータレイク管理者を個別に削除しようとすると、Lake Formation は無効なプリンシパルエラーをスローします。

エラー: "Caller is temporarily not authorized to perform this operation on resource..."

リダイレクトされた可能性のあるリソースリンクを使用してテーブルをクエリしようとしました。このエラーを回避するには、アプリケーションが QuerySessionContext プロトコルを採用する必要があります。詳細については、Lake Formation 「アプリケーション統合 API オペレーションのワークフロー」を参照してください。それ以外の場合は、24 時間待ってからクエリを再度実行してください。

クロスアカウントアクセスのトラブルシューティング

この情報を使用して、クロスアカウントアクセス問題の診断と修正に役立ててください。

トピック

- クロスアカウント Lake Formation 許可を付与しましたが、受領者がリソースを表示できません
- 受領者アカウントのプリンシパルは、Data Catalog リソースを表示することはできますが、基盤となるデータにはアクセスできません。
- エラー: AWS RAM リソース共有の招待を受け入れるときに、「発信者が承認されなかったため関連付けに失敗しました」
- エラー:「Not authorized to grant permissions for the resource」(リソースの許可を付与する権限がありません)
- エラー: AWS 「組織情報を取得するためのアクセスが拒否されました」
- エラー:「Organization <organization-ID> not found」(組織 <organization-ID> が見つかりません)
- エラー:「Insufficient Lake Formation permissions: Illegal combination」(Lake Formation 許可が不十分です: 不正な組み合わせ)
- 外部アカウントへのリクエストを許可/取り消ししたときに発生する ConcurrentModificationException
- Amazon EMR を使用して、クロスアカウント経由で共有されたデータにアクセスする際のエラー

クロスアカウント Lake Formation 許可を付与しましたが、受領者がリソースを表示できません

- 受領者アカウントのユーザーはデータレイク管理者ですか。共有時にリソースを表示できるのは、データレイク管理者のみです。
- 名前付きリソース方式を使用して組織外のアカウントとの共有を行っていますか。その場合、受信者アカウントのデータレイク管理者は AWS Resource Access Manager () でリソース共有の招待を受け入れる必要がありますAWS RAM。

  詳細については、「the section called “ AWS RAM リソース共有の招待の承諾”」を参照してください。
- AWS Glue でアカウントレベルの (Data Catalog) リソースポリシーを使用していますか。使用しているならば、名前付きリソース方式を使用する場合、 AWS RAM がユーザーに代わってポリシーを共有することを認可する特別なステートメントをポリシーに含める必要があります。

  詳細については、「the section called “AWS Glue と Lake Formation の両方を使用したクロスアカウント許可の管理”」を参照してください。
- クロスアカウントアクセスを付与するために必要な AWS Identity and Access Management (IAM) アクセス許可はありますか?

詳細については、「the section called “前提条件”」を参照してください。

- 許可を付与したリソースには、IAMAllowedPrincipals グループに付与された Lake Formation 許可がない必要があります。
- アカウントレベルポリシーに、リソースに対する deny ステートメントがありますか。

受領者アカウントのプリンシパルは、Data Catalog リソースを表示することはできますが、基盤となるデータにはアクセスできません。

受信者アカウントのプリンシパルには、必要な AWS Identity and Access Management (IAM) アクセス許可が必要です。詳細については、「共有テーブルの基盤となるデータへのアクセス」を参照してください。

エラー: AWS RAM リソース共有の招待を受け入れるときに、「発信者が承認されなかったため関連付けに失敗しました」

リソースへのアクセス権を別のアカウントに付与した後で、受領側アカウントがリソース共有招待を承諾しようとすると、アクションが失敗します。

```
$ aws ram get-resource-share-associations --association-type PRINCIPAL --resource-
share-arns arn:aws:ram:aws-region:444444444444:resource-share/e1d1f4ba-xxxx-xxxx-xxxx-
xxxxxxxx5d8d
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:aws-region:444444444444:resource-share/
e1d1f4ba-xxxx-xxxx-xxxx-xxxxxxxx5d8d
",
            "resourceShareName": "LakeFormation-MMCC0XQBH3Y",
            "associatedEntity": "5815803XXXXX",
            "associationType": "PRINCIPAL",
            "status": "FAILED",
            "statusMessage": "Association failed because the caller was not
 authorized.",
            "creationTime": "2021-07-12T02:20:10.267000+00:00",
            "lastUpdatedTime": "2021-07-12T02:20:51.830000+00:00",
            "external": true
        }
    ]
```

```
}
```

このエラーは、受領側アカウントがリソース共有招待を承諾するときに AWS Glue によって `glue:PutResourcePolicy` が呼び出されるために発生します。この問題を解決するには、プロデューサー/付与者アカウントによって使用される、引き受けられたロールによる `glue:PutResourcePolicy` アクションを許可します。

エラー:「Not authorized to grant permissions for the resource」(リソースの許可を付与する権限がありません)

別のアカウントが所有するデータベースまたはテーブルに対するクロスアカウント許可の付与が試行されました。データベースまたはテーブルがアカウントと共有されている場合、データレイク管理者としてこれらに対する許可を付与できるのは、アカウント内のユーザーのみです。

エラー: AWS 「組織情報を取得するためのアクセスが拒否されました」

アカウントは AWS Organizations 管理アカウントであり、アカウントの組織単位などの組織情報を取得するために必要なアクセス許可がありません。

詳細については、「Required permissions for cross-account grants」を参照してください。

エラー:「Organization <organization-ID> not found」(組織 <organization-ID> が見つかりません)

組織とのリソースの共有が試行されましたが、組織との共有が有効になっていません。組織とのリソース共有を有効にしてください。

詳細については、AWS RAM 「 ユーザーガイド」のAWS 「組織との共有を有効にする」を参照してください。

エラー:「Insufficient Lake Formation permissions: Illegal combination」(Lake Formation 許可が不十分です: 不正な組み合わせ)

リソースの `IAMAllowedPrincipals` グループに Lake Formation 許可が付与されているときに、ユーザーが Data Catalog リソースを共有しました。ユーザーは、リソースを共有する前に `IAMAllowedPrincipals` からすべての Lake Formation 許可を取り消す必要があります。

外部アカウントへのリクエストを許可/取り消ししたときに発生する ConcurrentModificationException

ユーザーが LF タグポリシーのプリンシパルに対する許可リクエストを複数同時に許可または取り消すと、Lake Formation は ConcurrentModificationException をスローします。ユーザーはこの例外を捕捉し、失敗した許可/取り消しリクエストを再試行する必要があります。バッチバージョンの `GrantPermissions`/`RevokePermissions` API オペレーション (BatchGrantPermissions および BatchRevokePermissions) を使用すると、同時許可/取り消しリクエストの数を減らすことで、この問題はある程度緩和されます。

Amazon EMR を使用して、クロスアカウント経由で共有されたデータにアクセスする際のエラー

Amazon EMR を使用して他のアカウントから共有されているデータにアクセスすると、一部の Spark ライブラリは `Glue:GetUserDefinedFunctions` API オペレーションの呼び出しを試みます。 AWS RAM 管理アクセス許可のバージョン 1 および 2 はこのアクションをサポートしていないため、次のエラーメッセージが表示されます。

```
"ERROR: User: arn:aws:sts::012345678901:assumed-role/my-
spark-role/i-06ab8c2b59299508a is not authorized to perform:
glue:GetUserDefinedFunctions on resource: arn:exampleCatalogResource
because no resource-based policy allows the glue:GetUserDefinedFunctions
action"
```

このエラーを解決するには、リソース共有を作成したデータレイク管理者が、リソース共有にアタッチされた AWS RAM マネージドアクセス許可を更新する必要があります。 AWS RAM マネージドアクセス許可のバージョン 3 では、プリンシパルが `glue:GetUserDefinedFunctions` アクションを実行できます。

新しいリソース共有を作成すると、Lake Formation はデフォルトで AWS RAM マネージドアクセス許可の最新バージョンを適用し、ユーザーによるアクションは必要ありません。既存のリソース共有のクロスアカウントデータアクセスを有効にするには、 AWS RAM マネージドアクセス許可をバージョン 3 に更新する必要があります。

で共有されているリソースに割り当てられた AWS RAM アクセス許可を表示できます AWS RAM。バージョン 3 には次のアクセス許可が含まれています。

```
Databases
```

```
  AWSRAMPermissionGlueDatabaseReadWriteForCatalog
  AWSRAMPermissionGlueDatabaseReadWrite

Tables
  AWSRAMPermissionGlueTableReadWriteForCatalog
  AWSRAMPermissionGlueTableReadWriteForDatabase

AllTables
  AWSRAMPermissionGlueAllTablesReadWriteForCatalog
  AWSRAMPermissionGlueAllTablesReadWriteForDatabase
```

既存のリソース共有の AWS RAM マネージドアクセス許可バージョンを更新するには

ユーザー (データレイク管理者) は、 AWS RAM ユーザーガイドの手順に従ってAWS RAM 管理アクセス許可を新しいバージョンに更新するか、リソースタイプの既存のアクセス許可をすべて取り消して再付与することができます。アクセス許可を取り消すと、 は AWS RAM リソースタイプに関連付けられたリソース共有 AWS RAM を削除します。アクセス許可を再付与すると、 AWS RAM は AWS RAM マネージドアクセス許可の最新バージョンをアタッチした新しいリソース共有を作成します。

ブループリントとワークフローのトラブルシューティング

この情報を使用して、ブループリントとワークフローの問題の診断と修正に役立ててください。

トピック

- ブループリントが「User: <user-ARN> is not authorized to perform: iam:PassRole on resource: <role-ARN>」(ユーザー: <user-ARN> にはリソース: <role-ARN> で iam:PassRole を実行する許可がありません) エラーで失敗しました
- ワークフローが「User: <user-ARN> is not authorized to perform: iam:PassRole on resource: <role-ARN>」(ユーザー: <user-ARN> にはリソース: <role-ARN> で iam:PassRole を実行する許可がありません) エラーで失敗しました
- ワークフローのクローラが「Resource does not exist or requester is not authorized to access requested permissions」(リソースが存在しないかリクエストされた認可にアクセスする権限がリクエスト元にありません) エラーで失敗しました
- ワークフローのクローラが「An error occurred (AccessDeniedException) when calling the CreateTable operation...」(CreateTable 操作の呼び出し時にエラーが発生しました (AccessDeniedException)) で失敗しました

ブループリントが「User: <user-ARN> is not authorized to perform: iam:PassRole on resource: <role-ARN>」(ユーザー: <user-ARN> にはリソース: <role-ARN> で iam:PassRole を実行する許可がありません) エラーで失敗しました

選択されたロールを渡すために十分な許可を持たないユーザーによって、ブループリントの作成が試行されました。

ロールを渡すことができるようにユーザーの IAM ポリシーを更新するか、必要な PassRole 許可を持つ異なるロールを選択することをユーザーに依頼してください。

詳細については、「the section called “Lake Formation のペルソナと IAM 許可のリファレンス”」を参照してください。

ワークフローが「User: <user-ARN> is not authorized to perform: iam:PassRole on resource: <role-ARN>」(ユーザー: <user-ARN> にはリソース: <role-ARN> で iam:PassRole を実行する許可がありません) エラーで失敗しました

ワークフローに指定したロールに、ロールがそれ自体を渡すことを許可するインラインポリシーがありませんでした。

詳細については、「the section called “(オプション) ワークフロー用の IAM ロールを作成する”」を参照してください。

ワークフローのクローラが「Resource does not exist or requester is not authorized to access requested permissions」(リソースが存在しないかリクエストされた認可にアクセスする権限がリクエスト元にありません) エラーで失敗しました

原因の 1 つとして、渡されたロールがターゲットデータベースにテーブルを作成するために十分な許可を持っていなかったことが考えられます。データベースに対する CREATE_TABLE 許可をロールに付与してください。

ワークフローのクローラが「An error occurred (AccessDeniedException) when calling the CreateTable operation...」(CreateTable 操作の呼び出し時にエラーが発生しました (AccessDeniedException)) で失敗しました

原因の 1 つとして、ワークフローロールがターゲットストレージロケーションに対するデータロケーション許可を持っていなかったことが考えられます。データロケーション許可をロールに付与してください。

詳細については、「the section called “DATA_LOCATION_ACCESS”」を参照してください。

の既知の問題 AWS Lake Formation

これらの既知の問題を確認します AWS Lake Formation。

トピック

- テーブルメタデータのフィルタリングの制限
- 除外された列の名前変更に関する問題
- CSV テーブルの列の削除に関する問題
- テーブルパーティションを共通パスの下に追加する必要性
- ワークフロー作成時におけるデータベースの作成に関する問題
- ユーザーの削除後での再作成に関する問題
- Data Catalog API 操作が IsRegisteredWithLakeFormation パラメータの値を更新しない
- Lake Formation オペレーションは AWS Glue スキーマレジストリをサポートしていません

テーブルメタデータのフィルタリングの制限

AWS Lake Formation 列レベルのアクセス許可を使用して、テーブル内の特定の列へのアクセスを制限できます。ユーザーがコンソールや `glue:GetTable` のような API を使用してテーブルに関するメタデータを取得する場合、テーブルオブジェクトの列リストには、ユーザーがアクセスできるフィールドのみが含まれます。このメタデータフィルタリングの制限を理解しておくことが重要です。

Lake Formation は、統合サービスが列の許可に関するメタデータを利用できるようにしますが、クエリ応答内の列の実際のフィルタリングは統合サービスの責任になります。Amazon Athena、Amazon Redshift Spectrum、および Amazon EMR などの列レベルのフィルタリングをサポートする Lake Formation クライアントは、Lake Formation に登録された列の許可に基づいてデー

タをフィルタリングします。ユーザーが、アクセス権を持つべきではないデータを読み取ることはできません。現在、AWS Glue ETL は列フィルタリングをサポートしていません。

> **Note**
>
> EMR クラスターは、 AWSが完全に管理しているわけではありません。このため、データへの不正アクセスを回避するためのクラスターの適切なセキュア化は、EMR 管理者の責任になります。

特定のアプリケーションまたはフォーマットでは、列の名前やタイプなどの追加のメタデータが、テーブルのプロパティとして `Parameters` マップに保存される場合があります。これらのプロパティは変更されずに返され、いずれかの列に対して `SELECT` 許可を持っていれば、どのユーザーでもアクセスすることができます。

例えば、Avro SerDe は `avro.schema.literal` というテーブルプロパティにテーブルスキーマの JSON 表現を保存し、このテーブルにアクセスできるすべてのユーザーが利用できます。機密情報をテーブルプロパティに保存することは避け、ユーザーが Avro 形式のテーブルの完全なスキーマを把握できることに留意することが推奨されます。この制限は、テーブルに関するメタデータに固有のものです。

AWS Lake Formation 呼び出し元にテーブル内のすべての列に対するSELECTアクセス許可がない場合、 は、 `glue:GetTable`または同様のリクエストに応答`spark.sql.sources.schema`するときに で始まるテーブルプロパティを削除します。これは、ユーザーが Apache Spark で作成されたテーブルに関する追加のメタデータにアクセスできないようにします。Apache Spark アプリケーションは、Amazon EMR で実行しても引き続きこれらのテーブルを読み取ることができますが、特定の最適化が適用されない場合があり、大文字と小文字を区別する列名はサポートされません。ユーザーがテーブル内のすべての列にアクセスできる場合、Lake Formation は、変更されていないテーブルをすべてのテーブルプロパティと共に返します。

除外された列の名前変更に関する問題

列レベルの許可を使用して列を除外してから列の名前を変更すると、その列は `SELECT *` などのクエリから除外されなくなります。

CSV テーブルの列の削除に関する問題

CSV 形式で Data Catalog のテーブルを作成した後でスキーマから列を削除すると、クエリが誤ったデータを返し、列レベルの許可が守られない場合があります。

回避方法: その代わりに新しいテーブルを作成します。

テーブルパーティションを共通パスの下に追加する必要性

Lake Formation は、テーブルのすべてのパーティションが、テーブルの [location] (ロケーション) フィールドに設定されている共通のパスの下にあることを期待します。これは、クローラを使用してカタログにパーティションを追加する場合は問題なく機能しますが、パーティションを手動で追加し、これらのパーティションが親テーブルに設定されたロケーションの下にない場合はデータアクセスが機能しません。

ワークフロー作成時におけるデータベースの作成に関する問題

Lake Formation コンソールを使用してブループリントからワークフローを作成するときは、ターゲットデータベースが存在しなければ、それを作成することができます。これを実行するとき、作成されるデータベースに対する CREATE_TABLE 許可を取得するのは、サインインしているユーザーです。しかし、ワークフローが生成するクローラは、テーブルの作成試行時にワークフローのロールを引き受けます。このロールにはデータベースに対する CREATE_TABLE 許可がないことから、テーブルの作成は失敗します。

回避方法: ワークフローのセットアップ中にコンソールからデータベースを作成する場合は、ワークフローを実行する前に、作成したばかりのデータベースに対する CREATE_TABLE 許可をワークフローに関連付けられているロールに付与する必要があります。

ユーザーの削除後での再作成に関する問題

以下のシナリオは、lakeformation:ListPermissions によって返される誤った Lake Formation 許可の原因になります。

1. ユーザーを作成し、Lake Formation 許可を付与。
2. ユーザーを削除。
3. 同じ名前のユーザーを再度作成。

ListPermissions は、古いユーザー向けのエントリと、新しいユーザー向けのエントリの 2 つのエントリを返します。古いユーザーに付与された許可を取り消そうとすると、それらの許可は新しいユーザーからも取り消されます。

Data Catalog API 操作が **IsRegisteredWithLakeFormation** パラメータの値を更新しない

GetTables および SearchTables などの Data Catalog API 操作が IsRegisteredWithLakeFormation パラメータの値を更新せず、デフォルト値の false を返すという既知の制限があります。IsRegisteredWithLakeFormation パラメータの正しい値を表示するには、GetTable API を使用することが推奨されます。

Lake Formation オペレーションは AWS Glue スキーマレジストリをサポートしていません

Lake Formation オペレーションは、スキーマレジスターで使用する StorageDescriptor SchemaReferenceの を含む AWS Glue テーブルをサポートしていません。

エラーメッセージを更新しました

AWS Lake Formation は、セキュリティとコンプライアンスの目標を達成するために、以下の API オペレーションのリソース固有の例外を一般的なEntityNotFoundエラーメッセージに更新しました。

- RevokePermissions
- GrantPermissions
- GetResourceLFTags
- GetTable
- GetDatabase

AWS Lake Formation API

> **Note**
>
> AWS Lake Formation サービスの更新された API リファレンスが利用可能になりました。

目次

- 許可 API
  - 操作
  - データ型
- データレイク設定 API
  - 操作
  - データ型
- IAM アイデンティティセンターの統合 API
  - 操作
  - データ型
- ハイブリッドアクセスモード API
  - 操作
  - データ型
- 認証情報供給 API
  - 操作
  - データ型
- API のタグ付け
  - 操作
  - データ型
- データフィルター API
  - 操作
  - データタイプ
- 一般的なデータ型
  - ErrorDetail 構造

- 文字列パターン

許可 API

「アクセス許可 API」セクションは、AWS Lake Formation でのアクセス許可の付与と取り消しに必要なオペレーションとデータ型について説明します。すべての AWS Lake Formation API オペレーションとデータ型については、「Lake Formation API Reference Guide」(Lake Formation API リファレンスガイド) を参照してください。

操作

- GrantPermissions
- RevokePermissions
- BatchGrantPermissions
- BatchRevokePermissions
- GetEffectivePermissionsForPath
- ListPermissions
- GetDataLakePrincipal

データ型

- リソース
- DatabaseResource
- TableResource
- TableWithColumnsResource
- DataCellsFilterResourcee
- DataLocationResource
- DataLakePrincipal
- PrincipalPermissions
- PrincipalResourcePermissions
- DetailsMap

- ColumnWildcard
- BatchPermissionsRequestEntry
- BatchPermissionsFailureEntry

データレイク設定 API

このセクションには、データレイク管理者を管理するためのデータレイク設定 API オペレーションとデータ型が含まれています。

操作

- GetDataLakeSettings
- PutDataLakeSettings

データ型

- DataLakeSettings

IAM アイデンティティセンターの統合 API

このセクションでは、Lake Formation と IAM アイデンティティセンターの統合を作成および管理するための操作について説明しています。

操作

- CreateLakeFormationIdentityCenterConfiguration
- DeleteLakeFormationIdentityCenterConfiguration
- DescribeLakeFormationIdentityCenterConfiguration
- UpdateLakeFormationIdentityCenterConfiguration

データ型

- ExternalFilteringConfiguration

ハイブリッドアクセスモード API

「ハイブリッドアクセスモード API」セクションでは、AWS Lake Formation でハイブリッドアクセスモードを設定するために必要なオペレーションとデータ型について説明します。すべての AWS Lake Formation API オペレーションとデータ型については、「Lake Formation API Reference Guide」(Lake Formation API リファレンスガイド) を参照してください。

操作

- CreateLakeFormationOptIn
- DeleteLakeFormationOptIn
- ListLakeFormationOptIns

データ型

- リソース
- DatabaseResource
- TableResource
- ResourceInfo
- LakeFormationOptInsInfo
- DataLocationResource

認証情報供給 API

「認証情報供給 API」セクションは、認証情報を供給し、データレイクリソースを登録および管理するための AWS Lake Formation サービスの使用に関連するオペレーションとデータ型について説明します。

操作

- RegisterResource
- DeregisterResource
- ListResources
- GetUnfilteredTableMetadata

- GetUnfilteredPartitionsMetadata
- GetTemporaryGluePartitionCredentials
- GetTemporaryGlueTableCredentials
- UpdateResource

データ型

- FilterCondition
- RowFilter
- ResourceInfo

API のタグ付け

「タグ付け API」セクションは、属性またはキーと値ペアのタグに対するアクセス許可モデルを定義する、認可戦略に関連するオペレーションとデータ型について説明します。

操作

- GetLFTagExpression
- ListLFTagExpressions
- DeleteLFTagExpression
- UpdateLFTagExpression
- CreateLFTagExpression
- AddLFTagsToResource
- RemoveLFTagsFromResource
- GetResourceLFTags
- ListLFTags
- CreateLFTag
- GetLFTag
- UpdateLFTag
- DeleteLFTag
- SearchTablesByLFTags

- SearchDatabasesByLFTags

データ型

- LFTagKeyResource
- LFTagPolicyResource
- TaggedTable
- TaggedDatabase
- LFTag
- LFTagPair
- LFTagError
- ColumnLFTag

データフィルター API

データフィルター API は、AWS Lake Formation でデータセルフィルターを管理する方法を記述します。

操作

- CreateDataCellsFilter
- DeleteDataCellsFilter
- ListDataCellsFilter
- GetDataCellsFilter
- UpdateDataCellsFilter

データタイプ

- DataCellsFilter
- RowFilter

一般的なデータ型

一般的なデータ型は、AWS Lake Formation でのその他の一般的なデータ型を記述します。

ErrorDetail 構造

エラーに関する詳細が含まれています。

フィールド

- `ErrorCode` – UTF-8 文字列。1 ~ 255 バイト長。Single-line string pattern に一致。

  このエラーに関連付けられたコード。
- `ErrorMessage` – 説明文字列。2,048 バイト長以下。URI address multi-line string pattern に一致。

  エラーを説明するメッセージ。

文字列パターン

API は、さまざまな文字列パラメータとメンバーの有効コンテンツを定義するために、以下の正規表現を使用します。

- 単一行文字列パターン – 「`[\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\t]*`」
- URI アドレスの複数行文字列パターン – 「`[\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\r\n\t]*`」
- カスタム文字列パターン 3 – 「」`^\w+\.\w+\.\w+$`
- カスタム文字列パターン 4 – 「」`^\w+\.\w+$`
- カスタム文字列パターン 5 – 「」`arn:aws:iam::[0-9]*:role/.*`
- カスタム文字列パターン 6 – 「」`arn:aws:iam::[0-9]*:user/.*`
- カスタム文字列パターン 7 – 「」`arn:aws:iam::[0-9]*:group/.*`
- カスタム文字列パターン 8 – 「」`arn:aws:iam::[0-9]*:saml-provider/.*`
- カスタム文字列パターン 9 – 「」`^([\p{L}\p{Z}\p{N}_.:\/=+\-@%]*)$`
- カスタム文字列パターン 10 – 「」`^([\p{L}\p{Z}\p{N}_.:\*\/=+\-@%]*)$`
- カスタム文字列パターン 11 – 「」`[\p{L}\p{N}\p{P}]*`

サポートされるリージョン

このセクションでは、Lake Formation でサポートされている AWS リージョン と機能について説明します。

一般提供

で AWS リージョン サポートされている については AWS Lake Formation、「 リージョンで利用可能な AWS サービスのリスト」を参照してください。

各リージョンの Lake Formation サービスエンドポイントと Lake Formation のサービスクォータのリストについては、「AWS Lake Formation エンドポイントとクォータ」を参照してください。

AWS GovCloud (US)

AWS GovCloud (US) リージョンと標準 の違いの概要については AWS リージョン、「 AWS Lake Formation の違い AWS GovCloud (US)」を参照してください。

トランザクションとストレージの最適化

Lake Formation の管理対象テーブル、トランザクションサポート、ストレージ最適化機能は、以下にあります AWS リージョン。

リージョン名	リージョンパラメータ	Endpoint
米国東部 (バージニア北部)	us-east-1	lakeformation.us-east-1.amazonaws.com lakeformation-fips.us-east-1.amazonaws.com
米国東部 (オハイオ)	us-east-2	lakeformation.us-east-2.amazonaws.com lakeformation-fips.us-east-2.amazonaws.com

リージョン名	リージョンパラメータ	Endpoint
米国西部 (オレゴン)	us-west-2	lakeformation.us-west-2.amazonaws.com lakeformation-fips.us-west-2.amazonaws.com
アジアパシフィック (ムンバイ)	ap-south-1	lakeformation.ap-south-1.amazonaws.com
アジアパシフィック (ソウル)	ap-northeast-2	lakeformation.ap-northeast-2.amazonaws.com
アジアパシフィック (シンガポール)	ap-southeast-1	lakeformation.ap-southeast-1.amazonaws.com
アジアパシフィック (シドニー)	ap-southeast-2	lakeformation.ap-southeast-2.amazonaws.com
アジアパシフィック (東京)	ap-northeast-1	lakeformation.ap-northeast-1.amazonaws.com
欧州 (フランクフルト)	eu-central-1	lakeformation.eu-central-1.amazonaws.com
欧州 (アイルランド)	eu-west-1	lakeformation.eu-west-1.amazonaws.com
欧州 (ロンドン)	eu-west-2	lakeformation.eu-west-2.amazonaws.com
欧州 (ストックホルム)	eu-north-1	lakeformation.eu-north-1.amazonaws.com

リージョン名	リージョンパラメータ	Endpoint
カナダ (中部)	ca-central-1	lakeformation.ca-central-1.amazonaws.com
南米 (サンパウロ)	sa-east-1	lakeformation.sa-east-1.amazonaws.com

のドキュメント履歴 AWS Lake Formation

次の表に、 ドキュメントの重要な変更点を示します AWS Lake Formation。

変更	説明	日付
属性ベースのアクセスコントロールに関するドキュメントを更新しました	Lake Formation では、データカタログリソースで一致する属性を持つプリンシパルにアクセス許可を付与できます。詳細については、「属性ベースのアクセス制御」を参照してください。	2025 年 4 月 17 日
Amazon S3 Tables と AWS Lake Formation および の統合 AWS Glue Data Catalog	S3 Tables を AWS Glue Data Catalog オブジェクトとして統合してカタログ化し、カタログを Lake Formation データの場所として登録できるようになりました。詳細については、「AWS Glue Data Catalogでの Amazon S3 Tables カタログの作成」を参照してください。	2025 年 3 月 13 日
Lake Formation のAWSLakeFormationCrossAccountManager ポリシーが更新されました。	Lake Formation は、StringLike 条件演算子を、IAM による ARN 形式チェックの実行を可能にする ArnLike 演算子に置き換えて、AWSLakeFormationCrossAccountManager ポリシーを強化しました。	2025 年 1 月 25 日
ポリシーの変更の更新	AWSLakeFormationDataAdmin ポリシーへの変更を文書化しました。	2024 年 12 月 3 日

マルチカタログの更新	AWS Glue Data Catalog を使用すると、フェデレーティッドカタログを作成し、Amazon S3 データレイクと Amazon Redshift データウェアハウス間でデータを統合したり、Amazon DynamoDB などの運用データベースや Snowflake、MySQL などのサードパーティーデータソースからのデータを統合したりできます。詳細については、「AWS Glue Data Catalogへのデータの取り込み」を参照してください。	2024 年 12 月 3 日
LF タグ式のドキュメントを更新	LF タグ式を保存し、再利用してデータカタログリソースに対するアクセス許可を付与できます。詳細については、「LF タグ式の管理」を参照してください。	2024 年 11 月 7 日
データカタログビューのドキュメントの更新	Amazon Athena と Amazon Redshift AWS Glue Data Catalog を使用して、DDLs に加えて AWS Glue APIs を使用して でビューを作成できます。詳細については、「データカタログビューの構築」を参照してください。	2024 年 8 月 7 日

監査可能な認証情報供給に関するドキュメントの追加	Lake Formation では、IAM アイデンティティセンターのユーザーのコンテキストを CloudTrail イベントに含めて、リソースにアクセスするユーザーを追跡することができます。詳細については、「CloudTrail 口グへの IAM アイデンティティセンターのユーザーコンテキストの追加」を参照してください。	2024 年 7 月 14 日
ポリシーの変更の更新	AWSLakeFormationCrossAccountManager および AWSLakeFormationDataAdmin ポリシーの変更 (ステートメント ID の追加と冗長なアクセス許可の削除) を文書化しました。	2024 年 3 月 14 日
Lake Formation のセットアップの更新	「AWS Lake Formationのセットアップ」セクションの手順を更新しました。	2024 年 2 月 7 日
ポリシーの変更の更新	サービスリンクロールのインラインポリシーに新しいアクセス許可を追加しました。詳細については、「Lake Formation のサービスリンクロールの使用」を参照してください。	2024 年 2 月 7 日
ポリシーの変更の更新	LakeFormationDataAccessServiceRolePolicy ポリシーの変更を文書化しました。	2024 年 2 月 2 日

Lake Formation の制限事項のまとめ	Lake Formation の制限事項と考慮事項をまとめたセクションを作成しました。詳細については、「Lake Formation の制限事項」を参照してください。	2023 年 12 月 15 日
Iceberg 圧縮に関するドキュメントを追加しました	Athena や Amazon EMR などの AWS 分析サービスや ETL AWS Glue ジョブによる読み取りパフォーマンスを向上させるために、 AWS Glue Data Catalog は Data Catalog の Iceberg テーブルに対してマネージド圧縮 (小さな Amazon S3 オブジェクトを大きなオブジェクトに圧縮するプロセス) を提供します。詳細については、「Iceberg テーブルの最適化」を参照してください。	2023 年 11 月 25 日
IAM アイデンティティセンターの統合に関するドキュメントが追加されました	IAM アイデンティティセンターの統合により、ユーザーとグループは Lake Formation 許可を適用するデータカタログリソースにアクセスできます。詳細については、「IAM アイデンティティセンターの統合」を参照してください。	2023 年 11 月 25 日

データカタログビューのドキュメントを追加しました	Amazon Athena または Amazon Redshift の SQL エディタを使用して、最大 10 個のテーブルを参照 AWS Glue Data Catalog するビューを に作成できます。詳細については、「ビューの作成」を参照してください。	2023 年 11 月 25 日
ポリシーの変更を更新	AWSLakeFormationCrossAccountManager ポリシーへの変更を文書化しました。	2023 年 10 月 25 日
ハイブリッドアクセスモードのドキュメントを追加しました	ハイブリッドアクセスモードでは、 AWS Glue Data Catalog内のデータベースとテーブルの Lake Formation 許可を柔軟かつ選択的に有効にできます。ハイブリッドアクセスモードを使用すると、他の既存のユーザーやワークロードのアクセス許可ポリシーを中断することなく、特定のユーザーのセットに Lake Formation 許可を設定できる増分パスが導入されました。詳細については、「ハイブリッドアクセスモード」を参照してください。	2023 年 9 月 26 日

Apache Iceberg テーブルを作成するためのドキュメントを追加しました	Amazon S3 にあるデータを使用して、 で Apache Parquet データ形式を使用する Apache Iceberg AWS Glue Data Catalog テーブルを作成できるようになりました。詳細については、「Iceberg テーブルの作成」を参照してください。	2023 年 8 月 16 日
クロスリージョンのデータアクセスに関するドキュメントを追加	Lake Formation は AWS 、リージョン間でのデータカタログテーブルのクエリをサポートしています。Athena、Amazon EMR を使用して他のリージョンからリージョンのデータにアクセスし、ソースデータベースとテーブルを指す他のリージョンにリソースリンクを作成することで AWS Glue ETL を実行できます。Amazon S3 データのメタデータを保存する外部メタストアに Data Catalog を接続し、 AWS Lake Formationを使用してデータアクセスのアクセス許可を安全に管理できます。詳細については、「クロスリージョンのテーブルアクセス」を参照してください。	2023 年 6 月 30 日
コンテンツを再編成	Lake Formation ユーザージャーニーに合わせて、ガイド内の章を再編成しました。	2023 年 5 月 15 日

HMS フェデレーションに関するドキュメントを追加	Amazon S3 データのメタデータを保存する外部メタストアに Data Catalog を接続し、AWS Lake Formationを使用してデータアクセスのアクセス許可を安全に管理できます。詳細については、「外部メタストアを使用するデータセットのアクセス許可の管理」を参照してください。	2023 年 4 月 15 日
Amazon Redshift データ共有に関するドキュメントを追加	Lake Formation のアクセス許可を使用して、Amazon Redshift からデータ共有内のデータを安全に管理できるようになりました。Lake Formation は、 を介したデータへのライセンスアクセスをサポートしています AWS Data Exchange。詳細については、「 でのデータ共有 AWS Lake Formation」を参照してください。	2022 年 11 月 30 日
プリンシパルとのクロスアカウントデータの直接共有のサポート	別のアカウントの IAM プリンシパルとデータを直接共有する方法についての情報を追加しました。詳細については、「AWS Lake Formationでのクロスアカウントデータ共有」を参照してください。	2022 年 11 月 10 日

TBAC を使用した AWS RAM 有効なデータ共有のサポート	クロスアカウント付与 AWS Resource Access Manager に使用するアクセス許可を Data Catalog に付与する LF-TBAC 方法に関する情報を追加しました。	2022 年 11 月 10 日
他のサービスとの連携に関するセクションを追加しました	Athena、Redshift Spectrum AWS Glue、Amazon EMR などの AWS サービスが Lake Formation を使用して、Lake Formation に登録されている Amazon S3 ロケーションのデータに安全にアクセスする方法に関する情報を追加しました。詳細については、「他の AWS サービスの使用」を参照してください。	2022 年 11 月 10 日
???	Amazon EMR を使用してクロスアカウントデータにアクセスする際の、エラーのトラブルシューティングに関する情報を追加しました。詳細については、「Amazon EMR を使用して、クロスアカウント経由で共有されたデータにアクセスする際のエラー」を参照してください。	2022 年 11 月 7 日
クロスアカウントのリソース共有の更新	Lake Formation でのクロスアカウントのリソース共有の仕組みに関する説明を追加しました。AWSLakeFormationCrossAccountManager ポリシーへの変更を文書化しました。	2022 年 5 月 6 日

新規チュートリアル	管理対象テーブルの作成、データレイクの保護、データレイクの共有に関する新しいチュートリアルを追加しました。詳細については、「入門チュートリアル」セクションを参照してください。	2022 年 4 月 20 日
新しい Lake Formation ランディングページ	Lake Formation ランディングページを更新して、Lake Formation を使用してデータレイクの構築、データの取り込み、データレイクの共有、データレイクの保護を行う方法についてステップバイステップの手順を提供するチュートリアルへのリンクを含めました。	2022 年 4 月 20 日
認証情報供給のサポート	認証情報供給に関する情報を追加しました。これにより、Lake Formation は、認証情報供給 API 操作を使用してサードパーティーサービスが Lake Formation と統合することを許可します。詳細については、「Lake Formation の認証情報供給の仕組み」を参照してください。	2022 年 2 月 28 日

管理対象テーブルと高度なデータフィルタリングのサポート	ACID トランザクション、自動データ圧縮、およびタイムトラベルクエリをサポートする管理対象テーブルに関する情報を追加しました。列レベルのセキュリティ、行レベルのセキュリティ、およびセルレベルのセキュリティをサポートするデータフィルターの作成に関する情報を追加しました。詳細については、「Lake Formation の管理対象テーブル」と「Lake Formation でのデータフィルタリングとセルレベルのセキュリティ」を参照してください。	2021 年 11 月 30 日
VPC インターフェイスエンドポイントのサポート	Lake Formation 用の Virtual Private Cloud (VPC) インターフェイスエンドポイントの作成に関する情報を追加し、VPC と Lake Formation 間の通信が AWS ネットワーク内で完全かつ安全に行われるようにしました。詳細については、「Lake Formation での VPC エンドポイントの使用」を参照してください。	2021 年 10 月 11 日

VPC エンドポイントポリシーのサポート	Lake Formation での仮想プライベートクラウド (VPC) エンドポイントポリシーのサポートに関する情報を追加しました。詳細については、「Lake Formation での VPC エンドポイントの使用」を参照してください。	2021 年 10 月 11 日
タグベースのアクセスコントロールのサポート	Lake Formation のタグベースのアクセスコントロールは、LF タグを使用することによって Data Catalog リソースと基盤となるデータへのアクセスを管理する、新しく、よりスケーラブルな方法を提供します。詳細については、「Lake Formation のタグベースのアクセスコントロール」を参照してください。	2021 年 5 月 7 日
Amazon EMR でのデータフィルタリングに関する新しいオプトイン要件。	Lake Formation によって管理されるデータの Amazon EMR によるフィルタリングを許可するためのオプトイン要件に関する情報を追加しました。詳細については、「Amazon EMR でのデータフィルタリングを許可する」を参照してください。	2020 年 10 月 9 日

Data Catalog データベースに対する完全なクロスアカウント許可の付与のサポート	CREATE_TABLE を含めた、AWS アカウント全体での Data Catalog データベースに対する完全な Lake Formation 許可の付与に関する情報を追加しました。詳細については、「Data Catalog データベースの共有」を参照してください。	2020 年 10 月 1 日
SAML を通じて認証する Amazon Athena ユーザーのサポート。	JDBC または ODBC ドライバー経由で接続し、Okta や Microsoft アクティブディレクトリフェデレーションサービス (AD FS) などの SAML ID プロバイダー経由で認証する Athena ユーザーのサポートに関する情報を追加しました。詳細については、「AWS サービスの Lake Formation との統合」を参照してください。	2020 年 9 月 30 日
暗号化された Data Catalog でのクロスアカウントアクセスのサポート	Data Catalog が暗号化されているときのクロスアカウント許可の付与に関する情報を追加しました。詳細については、「クロスアカウントアクセスの前提条件」を参照してください。	2020 年 7 月 30 日

データレイクへのクロスアカウントアクセスのサポート	外部 AWS アカウントと組織への Data Catalog データベースとテーブルに対する AWS Lake Formation アクセス許可の付与、および外部アカウントから共有される Data Catalog オブジェクトへのアクセスに関する情報を追加しました。詳細については、「クロスアカウントアクセス」を参照してください。	2020 年 7 月 7 日
Amazon Quick との統合	Quick Enterprise Edition ユーザーに Lake Formation アクセス許可を付与して、登録された Amazon S3 ロケーションにあるデータセットにアクセスできるようにする方法についての情報を追加しました。詳細については、「Data Catalog 許可の付与」を参照してください。	2020 年 6 月 29 日
設定と開始方法の章の更新	セットアップと使用開始に関する章を再編成し、改善しました。データレイク管理者の推奨 AWS Identity and Access Management (IAM) アクセス許可を更新しました。	2020 年 2 月 27 日

のサポート AWS Key Management Service	Lake Formation が AWS Key Management Service (AWS KMS) をサポートすることで、登録された Amazon Simple Storage Service (Amazon S3) ロケーションで暗号化されたデータを読み書きするための統合サービスの設定を簡素化する方法に関する情報を追加しました。暗号化されている Amazon S3 ロケーションを登録する方法に関する情報を追加しました AWS KMS keys。詳細については、「the section called “データレイクへの Amazon S3 ロケーションの追加”」を参照してください。	2020 年 2 月 27 日
ブループリントとデータレイク管理者 IAM ポリシーに対する更新	増分データベースブループリントの入力パラメータを明確にしました。データレイク管理者に必要な IAM ポリシーを更新しました。	2019 年 12 月 20 日
セキュリティに関する章の書き直しとアップグレードに関する章の改訂	セキュリティとアップグレードに関する章が改善されました。	2019 年 10 月 29 日
All (すべて) 許可の Super (スーパー) 許可への置き換え	セキュリティとアップグレードに関する章を更新して、`All` 許可の `Super` 許可への置き換えを反映しました。	2019 年 10 月 10 日

追加、訂正、および明確化	フィードバックに基づいて、追加、修正、および明確化を行いました。セキュリティに関する章を改訂しました。セキュリティとアップグレードに関する章を更新して、Everyone グループの IAMAllowedPrincipals グループへの置き換えを反映しました。	2019 年 9 月 11 日
新しいガイド	「AWS Lake Formation デベロッパーガイド」の初回リリースです。	2019 年 8 月 8 日

AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「AWS 用語集」を参照してください。

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。