AWS でのワークロード検出

実装ガイド

Mohsan Jaffery

Matthew Ball

Stefano Vozza

Connor Kirkpatrick

2020 年 *9* 月

最終更新日*: 2022* 年 *11* 月 *(*改訂を参照*)*

目次

はじめに

Amazon Web Services (AWS) クラウドのワークロードの監視活動は、運用上の健全性と効率性を維持するための鍵です。しかし、AWS リソースとワークロード間の関係性を追跡することは簡単ではありません。「AWS でのワークロード検出」ソリューションは、AWS クラウドのワークロードをアーキテクチャ図として自動的に可視化するツールです。このソリューションを使用して、AWS からのライブデータに基づいて詳細なワークロードの可視化を作成、カスタマイズ、共有できます。

このソリューションは、AWS アカウントと AWS リージョン全体の AWS リソースのインベントリを維持し、それらの関係性をマッピングして、ウェブユーザーインターフェイス (ウェブ UI) に表示します。「AWS でのワークロード検出」ソリューションは AWS マネジメントコンソール内のリソースへのリンクを提供します。これにより、リソースに変更を加えた場合の時間を節約できます。

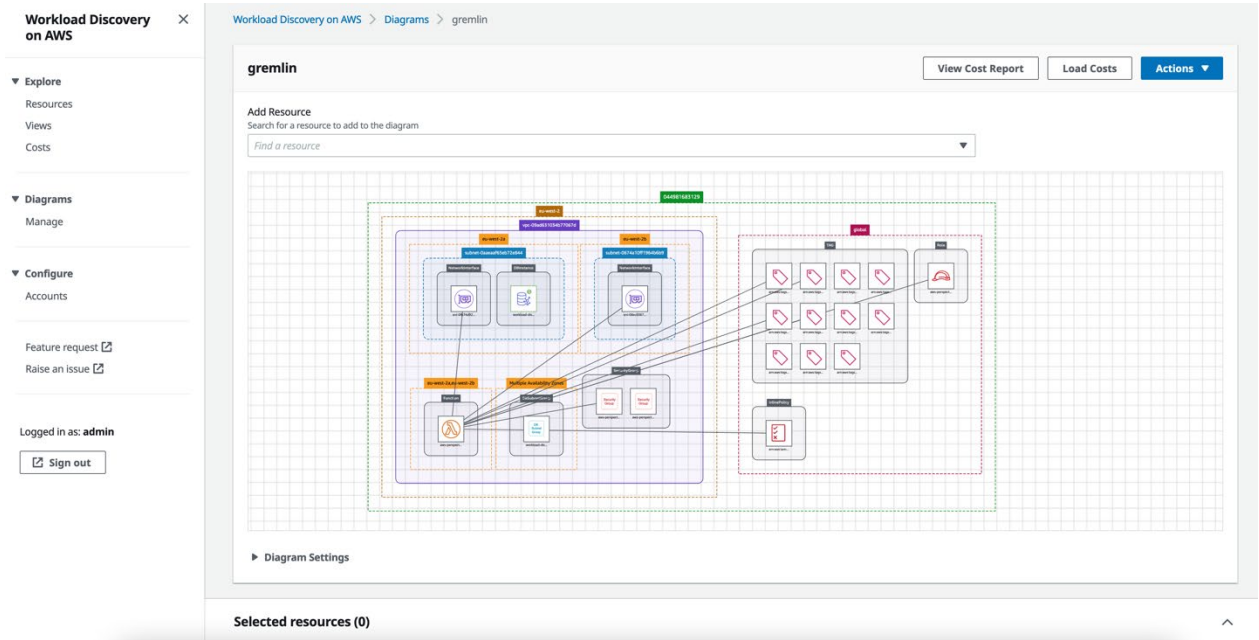

図 *1:*「*AWS* でのワークロード」ソリューションで作成したアーキテクチャ図のサンプル

この実装ガイドでは、AWS クラウドに「AWS でのワークロード検出」ソリューションをデプロイするためのアーキテクチャ上の考慮事項と設定手順について説明します。これには、セキュリティと可用性に関する AWS のベストプラクティスに沿って、必要な AWS のサービスを起動および設定する AWS CloudFormation テンプレートへのリンクが含まれています。

このガイドは、AWS クラウドの実践経験があるエンドユーザーを対象としています。

コスト

このソリューションの実行中にプロビジョニングされた AWS のサービスのコストは、お客様の負担となります。2022 年 11 月現在、米国東部 (バージニア北部) リージョンで、単一のインスタンスのデプロイオプションを使用してこのソリューションを実行するためのコストは、**1 時間あたり約 0.58 USD**、**1 か月あたり 425.19 USD** です。

「AWS でのワークロード検出」ソリューションを AWS クラウドで実行するためのコストは、デプロイ時に選択した設定によって異なることに注意してください。次の例は、米国東部 (バージニア北部) リージョンでの単一のインスタンスおよび複数インスタンスのデプロイ設定のコスト内訳を示しています。次の表でリストされている AWS のサービスは、月単位で請求されます。

コスト管理を容易にするために、AWS Cost Explorer を使用して予算を作成することを推奨しています。料金は変更される可能性があります。詳細については、このソリューションで使用される各 AWS サービスの料金表ウェブページを参照してください。

コストの例

オプション 1: 単一のインスタンスでデプロイ (デフォルト)

AWS CloudFormation テンプレートを使用してこのソリューションをデプロイする場合は、**OpensearchMultiAz** パラメータを `No` に設定すると、Amazon OpenSearch Service ドメイン用に単一のインスタンスがデプロイされ、**CreateNeptuneReplica** パラメータを `No` に設定すると、Amazon Neptune データストア用に単一のインスタンスをデプロイします。単一のインスタンスのデプロイオプションではコストが低くなりますが、アベイラビリティーゾーンに障害が発生した場合に「AWS でのワークロード検出」ソリューションの可用性が低下します。

AWS のサービス	インスタンスタイプ	時間あたりのコスト [USD]	1 か月あたりのコスト [USD]
Amazon Neptune	`db.r5.large`	0.348 USD	254.04 USD
Amazon OpenSearch Service	`m6g.large.search`	0.128 USD	93.44 USD
Amazon VPC (NAT Gateway)	N/A	0.090 USD	65.7 USD
AWS Config	N/A	リソースごとに 0.003 USD	リソースごとに 0.003 USD
Amazon ECS (AWS Fargate タスク)	N/A	0.02 USD	12.01 USD
	合計	**0.586 USD***	**425.19 USD***

オプション 2: 複数インスタンスでデプロイ

AWS CloudFormation テンプレートを使用してこのソリューションをデプロイする場合は、**OpensearchMultiAz** パラメータを `Yes` に設定すると、Amazon OpenSearch Service ドメイン用に 2 つのアベイラビリティーゾーンに 2 つのインスタンスがデプロイされ、**CreateNeptuneReplica** パラメータを `Yes` に設定すると、Amazon Neptune データストア用に 2 つのアベイラビリティーゾーンに 2 つのインスタンスがデプロイされます。複数インスタンスのデプロイオプションは実行コストが高くなりますが、アベイラビリティーゾーンに障害が発生した場合に「AWS でのワークロード検出」ソリューションの可用性が向上します。

AWS のサービス	インスタンスタイプ	時間あたりのコスト	1 か月あたりのコスト
Amazon Neptune	`db.r5.large`	0.696 USD	508.08 USD
Amazon OpenSearch Service	`m6g.large.search`	0.256 USD	186.88 USD
Amazon VPC (NAT Gateway)	N/A	0.090 USD	65.7 USD
AWS Config	N/A	リソースごとに 0.003 USD	リソースごとに 0.003 USD
Amazon ECS (AWS Fargate タスク)	N/A	0.02 USD	12.01 USD
	合計	**1.062 USD***	**772.67 USD***

* 最終的なコストは、AWS Config が検出したリソースの数によって異なります。表に記載されている金額に加えて、記録されたリソースアイテムにつき 0.003 USD が発生します。

> **重要:** Amazon Neptune と Amazon OpenSearch Service のコストは、デプロイ時に選択したインスタンスタイプによって異なります。

コスト管理を容易にするために、AWS Cost Explorer を使用して予算を作成することを推奨しています。料金は変更される可能性があります。詳細については、このソリューションで使用する AWS の各サービスの料金表のウェブページを参照してください。

アーキテクチャの概要

このソリューションをデフォルトのパラメータでデプロイすると、AWS クラウドに次の環境が構築されます。

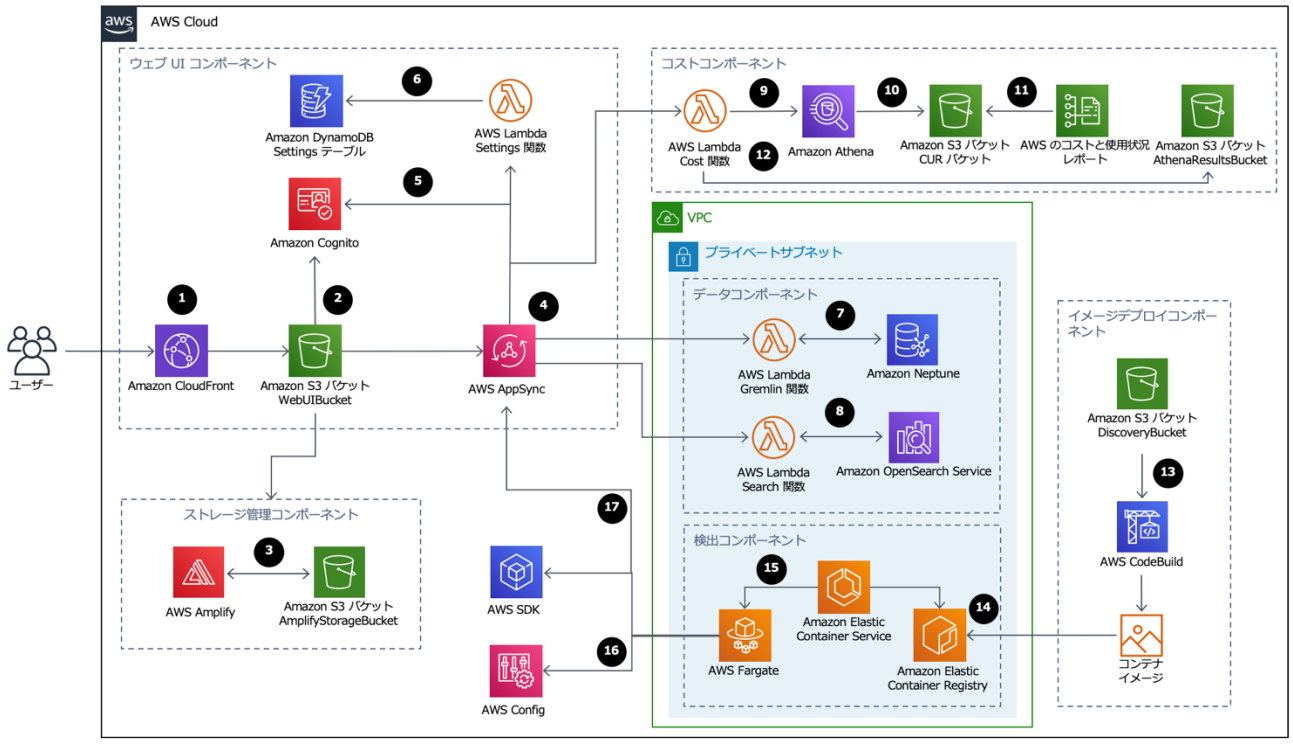

図 *2:「AWS でのワークロード検出」ソリューションのアーキテクチャ*

AWS CloudFormation テンプレートは、「AWS でのワークロード検出」ソリューションを AWS アカウントにデプロイします。次の概要では、このソリューションとともにデプロイされる 6 つのコンポーネントとそれらに関連する AWS のサービスについて説明します。各コンポーネントの詳細については、「ソリューションコンポーネント」セクションを参照してください。

1. Amazon CloudFront ディストリビューションからの各レスポンスに HTTP Strict-Transport-Security (HSTS) セキュリティヘッダを追加します。
2. Amazon Simple Storage Service (Amazon S3) バケットは、Amazon CloudFront を使用して配信されるウェブユーザーインターフェイス (ウェブ UI) をホストします。Amazon Cognito は、ウェブ UI へのユーザーアクセスを認証します。
3. AWS Amplify と Amazon S3 バケットは、ユーザープリファレンスと保存されたアーキテクチャ図を保持するストレージ管理コンポーネント用にデプロイされます。
4. AWS AppSync エンドポイントは、ウェブ UI コンポーネントがリソースの関係性のデータをリクエストしたり、コストをクエリしたり、新しい AWS リージョンをインポートしたり、環境設定を更新したりすることを可能にします。また、検出コンポーネントがこのソリューションのデータベースに永続的なデータを保存することもできます。
5. AWS AppSync は、Amazon Cognito によってプロビジョニングされる JSON ウェブトークン (JWT) を使用して、各リクエストを認証します。
6. `Settings` AWS Lambda 関数は、インポートされた AWS リージョンとその他の設定を Amazon DynamoDB に保持します。
7. データコンポーネントは、`Gremlin Resolver` AWS Lambda 関数を使用して、Amazon Neptune データベースからデータをクエリして返します。
8. データコンポーネントは、`Search Resolver` AWS Lambda 関数を使用して、Amazon OpenSearch Service ドメインに対するリソースデータのクエリと保存を行います。
9. `Cost` AWS Lambda 関数は、Amazon Athena を使用して、AWS Cost and Usage Report (AWS CUR) をクエリして、予想コストデータをウェブ UI に提供します。
10. Amazon Athena は AWS CUR 上でクエリを実行します。

11. AWS CUR は、レポートを `CostAndUsageReportBucket` Amazon S3 バケットに配信します。
12. `Cost` AWS Lambda 関数は、Amazon Athena の結果を `AthenaResultsBucket` Amazon S3 バケットに保存します。
13. AWS CodeBuild は、イメージデプロイコンポーネントで検出コンポーネントのコンテナイメージを作成します。
14. Amazon Elastic Container Registry (Amazon ECR) は、イメージデプロイコンポーネントによって提供される Docker イメージを保存します。
15. Amazon Elastic Container Service (Amazon ECS) は、AWS Fargate タスクを管理し、必要な構成を提供して、タスクを実行します。AWS Fargate は 15 分ごとにコンテナタスクを実行し、インベントリとリソースデータを更新します。
16. AWS Config と AWS SDK の呼び出しは、検出コンポーネントからインポーされた AWS リージョンからのリソースデータのインベントリを更新してから、その結果をデータコンポーネントに保存します。
17. AWS Fargate タスクは、AWS Config と AWS SDK の呼び出しの結果を、AppSync API への API コールを介して、Amazon Neptune データベースと Amazon OpenSearch Service ドメインに保存します。

ソリューションコンポーネント

認証メカニズム

「AWS でのワークロード検出」ソリューションでは、ウェブユーザーインターフェイス (ウェブ UI) と Amazon API Gateway 認証の両方に Amazon Cognito ユーザープール を使用します。認証されると、Amazon Cognito は JSON ウェブトークン (JWT) をウェブ UI に発行します。このトークンは、後続のすべての API リクエストで送信されます。有効な JWT が送信されていない場合は、API リクエストは失敗し、HTTP 403 Forbidden レスポンスを返します。

ウェブ UI とストレージ管理

ウェブ UI は React で開発され、ユーザーが「AWS でのワークロード検出」ソリューションを操作できるようにするフロントエンドコンソールを提供します。

Amazon CloudFront は、ウェブ UI へのすべての HTTP リクエストにセキュアなヘッダーを追加するよう設定されています。これにより、クロスサイトスクリプティング (XSS) などの攻撃から保護するセキュリティレイヤーが提供されます。

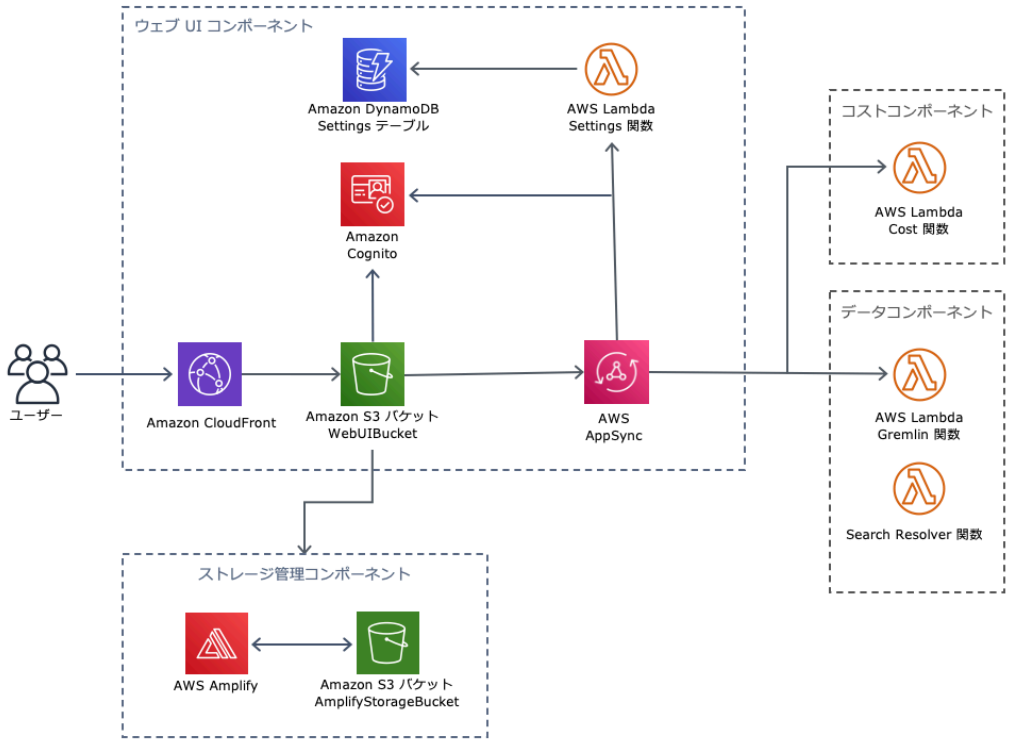

図 *3:*「*AWS* でのワークロード検出」ソリューションのウェブ *UI* とストレージ管理コンポーネント

ウェブ UI リソースは `WebUIBucket` Amazon S3 バケットでホストされ、Amazon CloudFront によって配信されます。AWS Amplify は、AWS AppSync および Amazon S3 への統合をシンプルにする抽象化レイヤーを提供します。Amazon Cognito は、ログインステージでユーザーを認証します。ログインに成功すると、Amazon Cognito からの認証レスポンスに JSON ウェブトークン (JWT) が発行されます。JWT は、後続のすべての API リクエストとともに送信する必要があります。JWT が送信されていない場合は、API リクエストは失敗し、HTTP 403 Forbidden レスポンスを返します。

AWS AppSync は、インポートされた AWS リージョンの管理など、「AWS でのワークロード検出」ソリューションで利用できる様々な設定とのやり取りを容易にするために使用されます。AWS AppSync は、`Settings` AWS Lambda 関数を使用して、新しい AWS リージョンのインポートなどのリクエストを処理します。

データコンポーネント

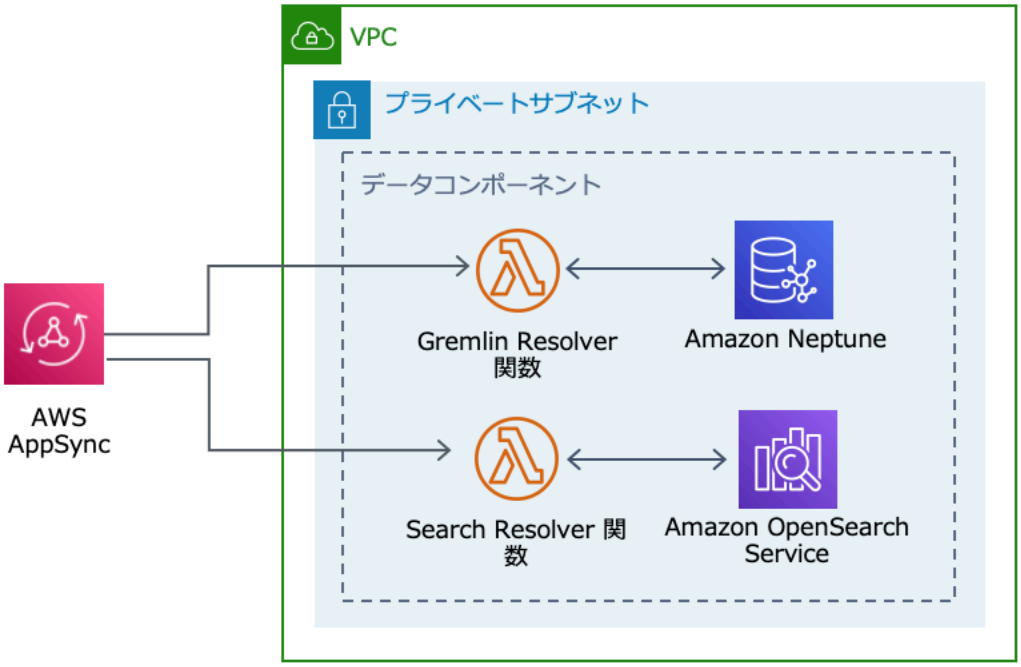

図 *4:「AWS でのワークロード検出」ソリューションのデータコンポーネント*

ウェブ UI は AppSync API にリクエストを送信します。これにより、`Gremlin Resolver` または `Search Resolver` の AWS Lambda 関数のいずれかが呼び出されます。これらの関数は Amazon Neptune または Amazon OpenSearch Service にクエリを実行して、提供されたリソースに関するデータを取得します。また、AWS AppSync は、AWS CUR からの予想コストデータのリクエストもサポートしています。

検出コンポーネントは AppSync API にリクエストを送信して、Amazon Neptune と Amazon OpenSearch Service のデータベースからデータを読み取って保持します。この API は、検出コンポ

ーネントの AWS Fargate タスクからリクエストを受け取り、データベースへのアクセス権限を持つ AWS Identity and Access Management (IAM) ロールによる認証を行います。

イメージデプロイコンポーネント

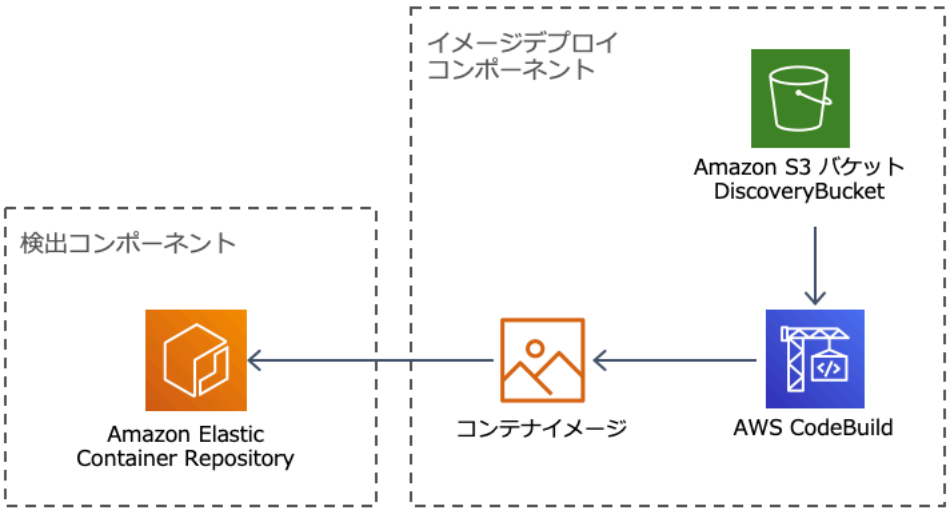

図 *5:「AWS でのワークロード検出」ソリューションのイメージデプロイコンポーネント*

イメージデプロイコンポーネントは、検出コンポーネントで使用されるコンテナイメージを作成します。コードは `DiscoveryBucket` Amazon S3 バケットでホストされ、コンテナイメージを構築して Amazon ECR にアップロードする AWS CodeBuild のジョブによってデプロイ時にダウンロードされます。

検出コンポーネント

検出コンポーネントは、「AWS でのワークロード検出」ソリューションのアーキテクチャの主要なデータ収集エレメントです。AWS Config へのクエリおよび AWS API の *describe* API コールの実行により、リソースのインベントリとリソース同士の関係性を維持します。

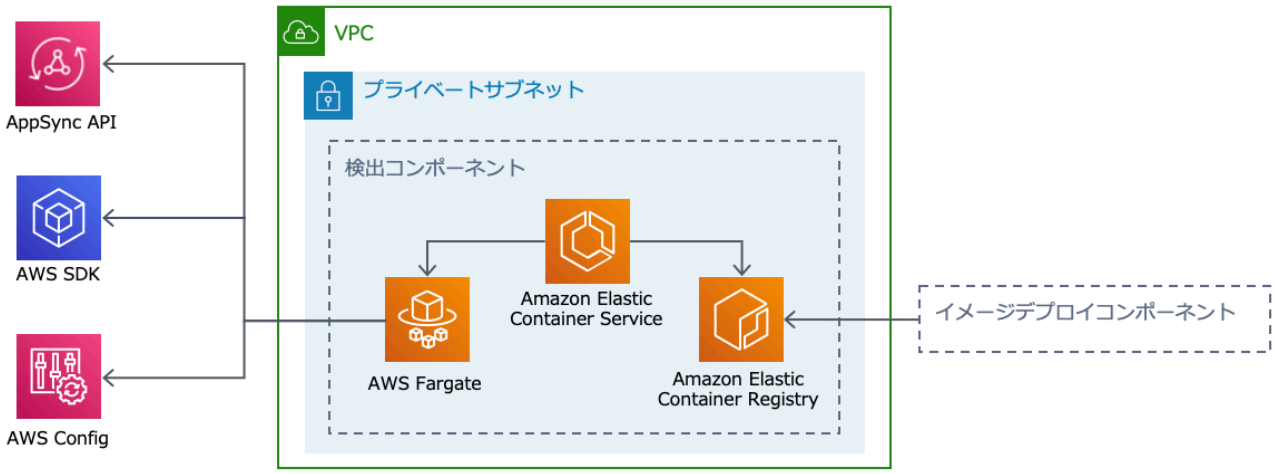

図 *6:*「*AWS* でのワークロード検出」ソリューションの検出コンポーネント

このソリューションでは、Amazon ECR からダウンロードされたコンテナイメージを使用して AWS Fargate タスクを実行するように Amazon ECS を設定します。AWS Fargate タスクは 15 分間隔で実行されるようにスケジュールされています。収集されたリソースの関係性のデータは、Amazon Neptune グラフデータベースと Amazon OpenSearch Service に保存されます。

検出コンポーネントのワークフローは、次の 3 つのステップで構成されています。

1. Amazon ECS は 15 分間隔で AWS Fargate タスクを実行します。
2. AWS Fargate タスクは、AWS Config、AWS API の *describe* API コール、Amazon Neptune データベースからリソースデータを収集します。
3. AWS Fargate タスクは、Amazon Neptune データベースに現在あるものと、AWS Config と *describe* API コールから受信したものとの差分を計算します。
4. AWS Fargate タスクは、AppSync API に送信して、検出されたリソースと関係性の変更を Amazon Neptune と Amazon OpenSearch Service に保持します。

コストコンポーネント

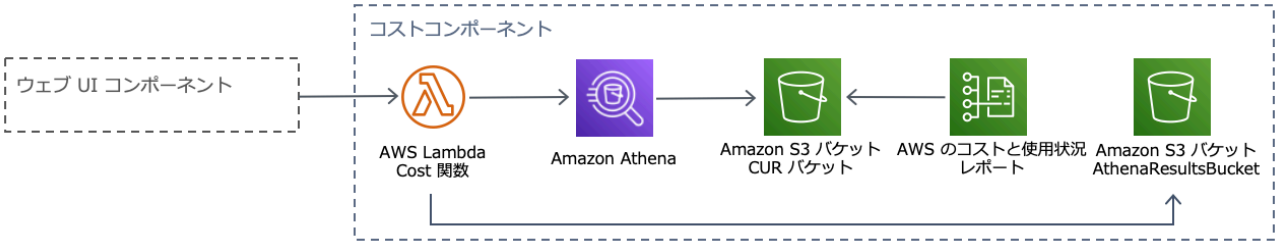

図 *7:*「*AWS* でのワークロード検出」ソリューションのコストコンポーネント

AWS CUR は AWS Billing and Cost Management で作成できます。これにより、Parquet 形式のファイルが `CostAndUsageReportBucket` Amazon S3 バケットにパブリッシュされます。ウェブ UI は、`Cost` AWS Lambda 関数を呼び出す AWS AppSync エンドポイントにリクエストを送信します。この関数は、AWS CUR から予想コスト情報を返す事前に定義されたクエリを Amazon Athena に送信します。

AWS CUR のサイズにより、Amazon Athena からのレスポンスが非常に大きくなる場合があります。このソリューションでは、`AthenaResultsBucket` Amazon S3 バケットに結果を保存し、その結果をウェブ UI にページ分割して戻します。このバケットに設定されたライフサイクルポリシーは、7 日以上経過したアイテムを削除します。

サポートされているリソース

「AWS でのワークロード検出」ソリューションが AWS アカウントと AWS リージョン内で検出できる AWS のリソースタイプの一覧については、「サポートされているリソース」を参照してください。

「AWS でのワークロード検出」ソリューションでのアーキテクチャ図の管理

「AWS でのワークロード検出」ソリューションで生成されるアーキテクチャ図は、作成、読み取り、更新、削除 (CRUD) 操作を実行できるウェブ UI を使用して保存できます。AWS Amplify の storage

API を使用して、「AWS でのワークロード検出」ソリューションは、アーキテクチャ図を Amazon S3 バケットに保存します。次の 2 つの権限が利用可能です。

- **All users** -「AWS でのワークロード検出」ソリューションのアーキテクチャ図をデプロイ内のユーザーに表示できるようにします。ユーザーは、これらの図をダウンロードして編集することができます。
- **You** -「AWS でのワークロード検出」ソリューションのアーキテクチャ図を作成者のみに表示されるようにします。他のユーザーには表示されません。

セキュリティ

AWS インフラストラクチャでシステムを構築する場合、セキュリティ上の責任はお客様と AWS の間で共有されます。この責任共有モデルでは、ホストオペレーティングシステム、仮想化レイヤー、サービスを運用する施設の物理的なセキュリティなどのコンポーネントを AWS が運用、管理、制御するため、お客様の運用上の負担を軽減します。AWS セキュリティの詳細については、「AWS セキュリティセンター」を参照してください。

「AWS でのワークロード検出」ソリューションは、セキュアを保つように設計および構成されています。次のような「AWS でのワークロード検出」ソリューションとそのコンポーネント部分のベストプラクティスが適用されています。

- アクセスは最小限の権限を付与するように設定され、可能な限り必要なリソースのみに限定されます。
- 保管中および転送中のデータは、専用のキー管理ストアである AWS Key Management Service (AWS KMS) に保存されているキーを使用して暗号化されます。
- 認証情報には有効期間が短く強力なパスワードポリシーを実装します。
- 必要に応じて、ロギング、トレース、およびバージョニングを有効にします。
- 適用可能な場合は、自動パッチ適用 (マイナーバージョン) とスナップショット作成を有効にします。

- ネットワークアクセスはデフォルトでプライベートになっており、Amazon Virtual Private Cloud (Amazon VPC) エンドポイントが使用可能な場合は有効にします。

リソースへのアクセス

IAM ロール

AWS Identity and Access Management (IAM) ロールにより、AWS クラウドのサービスとユーザーに対してアクセスポリシーとアクセス許可を詳細に割り当てることができます。「AWS でのワークロード検出」ソリューションを実行して、AWS アカウントのリソースを検出するには、複数のロールが必要です。

Amazon Cognito

Amazon Cognito は、「AWS でのワークロード検出」ソリューションに必要なコンポーネントへのアクセスを認証し、一時的な認証情報を提供します。

ネットワークアクセス

Amazon Virtual Private Cloud (Amazon VPC)

「AWS でのワークロード検出」ソリューションは Amazon VPC 内にデプロイされ、ベストプラクティスに従ってセキュリティと高可用性を実現します。詳細については、「VPC のセキュリティのベストプラクティス」を参照してください。VPC エンドポイントは、インターネットを経由しないサービス間の通信を可能とし、使用可能な場合に設定されます。

セキュリティグループは、「AWS でのワークロード検出」ソリューションの実行に必要なコンポーネント間のネットワークトラフィックを制御し分離するために使用されます。

デプロイが完了したら、セキュリティグループを確認し、必要に応じてアクセスをさらに制限することをお勧めします。

Amazon CloudFront

このソリューションでは、Amazon CloudFront によって配信される Amazon S3 バケットにホストされるウェブコンソールをデプロイします。この Amazon S3 バケットのコンテンツには、Amazon CloudFront を介してのみアクセスできます。オリジンアクセスアイデンティティ機能を使用してアクティブにします。詳細については、*Amazon CloudFront* 開発者ガイドの「オリジンアクセスアイデンティティを使用して Amazon S3 コンテンツへのアクセスを制限する」を参照してください。

追加のセキュリティ施策は、Amazon CloudFront が各ビューワーのレスポンスに HTTP セキュリティヘッダーを追加することでアクティブになります。詳細については、「Add HTTP security headers」を参照してください。

このソリューションでは、TLS v1.0 のみをサポートするデフォルトの Amazon CloudFront 証明書を使用しています。TLS v1.1 または TLS v1.2 を使用するには、デフォルトの Amazon CloudFront 証明書の代わりに独自の SSL 証明書を使用する必要があります。詳細については、「SSL/TLS 証明書を使用するように CloudFront ディストリビューションを設定する方法を教えてください」を参照してください。

アプリケーションの設定

AWS AppSync

「AWS でのワークロード検出」ソリューションの GraphQL API には、GraphQL 仕様に従って、AWS AppSync によってリクエスト検証が行われます。さらに、認証と認可は AWS IAM と Amazon Cognito を使用して実装されます。これは、ウェブ UI でユーザー認証に成功したときに、Amazon Cognito が発行する JSON ウェブトークン (JWT) によって実現されます。

AWS Lambda

デフォルトでは、AWS Lambda 関数は最新の安定したバージョンの言語ランタイムで設定されます。機密データやシークレットは記録されません。サービスの操作は、最小限必要な権限で実行されます。これらの権限を定義するロールは、関数間で共有されません。

Amazon OpenSearch Service

Amazon OpenSearch Service ドメインには、Amazon OpenSearch Service クラスターに対して行われた署名されていないリクエストを停止するために、アクセスを制限するアクセスポリシーが設定されています。これは単一の AWS Lambda 関数に制限されています。

Amazon OpenSearch Service クラスターは、ノード間の暗号化をアクティブにして構築されており、Amazon OpenSearch Service の既存のセキュリティ機能に追加のデータ保護のレイヤーを追加します。

設計に関する考慮事項

デプロイ用アカウントの作成

このソリューションのために作成された専用の AWS アカウントに「AWS でのワークロード検出」ソリューションをデプロイすることをお勧めします。このアプローチにより、「AWS でのワークロード検出」ソリューションは既存のワークロードから分離され、ユーザーの追加や新しい AWS リージョンのインポートなど、このソリューションを設定するための単一の環境を提供します。また、このソリューションの実行中に発生したコストを追跡しやすくなります。

「AWS でのワークロード検出」ソリューションがデプロイされると、プロビジョニング済みのアカウントから AWS リージョンをインポートできるようになります。

利用可能なデプロイリージョン

次の表は、「AWS でのワークロード検出」ソリューションでサポートしている AWS リージョンの一覧です。

リージョン ID	リージョン名
us-east-1	米国東部 (バージニア北部)
us-east-2	米国東部 (オハイオ)

リージョン ID	リージョン名
us-west-2	米国西部 (オレゴン)
ap-south-1	アジアパシフィック (ムンバイ) *
ap-northeast-2	アジアパシフィック (ソウル)
ap-southeast-1	アジアパシフィック (シンガポール)
ap-southeast-2	アジアパシフィック (シドニー)
ap-northeast-1	アジアパシフィック (東京)
ca-central-1	カナダ (中部)
eu-west-2	欧州 (ロンドン)
eu-central-1	欧州 (フランクフルト)
eu-west-1	欧州 (アイルランド)
eu-west-3	欧州 (パリ) **
eu-north-1	欧州 (ストックホルム)
sa-east-1	南米 (サンパウロ)

* デプロイ時に、OpensearchInstanceType パラメータに c6g.large.search を選択します。

** デプロイ時に、OpensearchInstanceType パラメータに m5.large.search を選択します。

AWS CloudFormation テンプレート

このソリューションでは、AWS CloudFormation を使用して、AWS クラウドへの「AWS でのワークロード検出」ソリューションのデプロイを自動化します。このソリューションには次の AWS CloudFormation テンプレートが含まれており、デプロイ前にダウンロード可能です。

テンプレートを表示

workload-discovery-on-aws.template: このテンプレートを使用して、ソリューションとすべての関連コンポーネントを起動します。デフォルト設定では、Amazon CloudFront、Amazon Simple Storage Service (Amazon S3)、AWS Lambda、Amazon Cognito、Amazon Athena、AWS AppSync、Amazon Neptune、Amazon OpenSearch Service、Amazon Elastic Container Service (Amazon ECS)、

AWS Fargate、AWS Config、Amazon Elastic Container Registry (Amazon ECR)、Amazon DynamoDB、AWS CodeBuild がデプロイされます。

> **注意:** このテンプレートは特定のニーズに合わせてカスタマイズできますが、変更を加えるとアップグレードプロセスに影響を与える可能性があります。

自動デプロイ

> **注意:** 既に「AWS でのワークロード検出」ソリューションをデプロイしていて、最新バージョンにアップグレードしたい場合は、「スタックの更新」を参照してください。

このソリューションを起動する前に、アーキテクチャ、設定、ネットワークセキュリティなどの、このガイドで説明している考慮事項を確認してください。このセクションの手順に従って、このソリューションを設定して AWS アカウントにデプロイします。

デプロイ時間：約 30 分

前提条件

デプロイパラメータの詳細を収集

「AWS でのワークロード検出」ソリューションをデプロイする前に、Amazon OpenSearch Service のサービスにリンクされたロールと AWS Config の設定詳細を確認します。

AWSServiceRoleForAmazonOpenSearchService ロールがあるか確認する

デプロイにより、Amazon Virtual Private Cloud (Amazon VPC) 内に Amazon OpenSearch Service クラスターが作成されます。このテンプレートでは、サービスにリンクされたロールを使用して Amazon OpenSearch Service クラスターを作成します。ただし、AWS アカウントにロールが作成済みである場合は、既存のロールを使用します。

このロールが既にあるか確認するには、次の手順を実行します。

1. このソリューションをデプロイする予定の AWS アカウントの AWS Identity and Access Management (IAM) コンソールにサインインします。
2. メニューの下にある **IAM の検索**ボックスで、`AWSServiceRoleForAmazonOpenSearchService` を検索します。

検索でロールが返された場合は、スタックの起動時に **CreateOpensearchServiceRole** パラメータで `No` を選択します。

AWS Config が設定されていることを確認する

「AWS でのワークロード検出」ソリューションでは、AWS Config を使用して、リソース設定の大半を収集しています。このソリューションをデプロイしたり、新しい AWS リージョンをインポートしたりする場合は、AWS Config が既にセットアップされ、期待どおりに動作しているかどうかを確認する必要があります。**AlreadyHaveConfigSetup** AWS CloudFormation パラメータは、AWS Config を設定するかどうかを「AWS でのワークロード検出」ソリューションに通知します。

次のスニペットは、AWS CLI コマンドリファレンスからの抜粋です。「AWS でのワークロード検出」ソリューションをデプロイするか、「AWS でのワークロード検出」ソリューションにインポートする AWS リージョンでこのコマンドを実行します。

```
aws configservice get-status
```

出力結果:

```
Configuration Recorders:

name: default
recorder: ON
last status: SUCCESS

Delivery Channels:

name: default
last stream delivery status: SUCCESS
last history delivery status: SUCCESS
last snapshot delivery status: SUCCESS
```

上記の出力結果と同様のレスポンスを受信した場合は、その AWS リージョンで Configuration Recorder と Delivery Channel が実行されています。**AlreadyHaveConfigSetup** AWS CloudFormation パラメーターで `Yes` を選択してください。

AWS CloudFormation StackSets を設定している場合は、AWS Config が既に設定されている AWS リージョンのバッチにこの AWS リージョンを含める必要があります。

自身のアカウントで *AWS Config* の詳細を確認する

デプロイ時に AWS Config の設定が行われます。デプロイ予定の AWS アカウントで AWS Config を既に使用している、または「AWS でのワークロード検出」ソリューションで検出可能にしている場合は、このソリューションをデプロイするときに関連するパラメータを選択します。さらに、デプロイを成功させるには、AWS Config がスキャンするリソースを制限していないことを確認してください。

現在の AWS Config 設定を確認するには、次の手順を実行します。

1. AWS Config コンソールにサインインします。
2. [**設定**] を選択し、「**このリージョンでサポートされているすべてのリソースを記録します**」ボックスと「**グローバルリソース (AWS IAM リソースなど) を含める**」ボックスがオンになっていることを確認します。

スタックの起動

> **重要:** このソリューションには、匿名の運用メトリクスを AWS に送信するオプションが含まれています。当社はこのデータを使用して、ユーザーがこのソリューション、関連サービスおよび製品をどのように使用しているかをよりよく理解し、提供するサービスや製品の改善に役立てます。AWS は、このアンケートを通じて収集されたデータを所有します。データ収集には、AWS プライバシーポリシーが適用されます。
>
> この機能を無効にするには、デプロイ時に OptOutOfSendingAnonymousUsageMetrics パラメータを `Yes` に設定してください。

この自動化された AWS CloudFormation テンプレートは、AWS クラウドに「AWS でのワークロード検出」ソリューションをデプロイします。スタックを起動する前に、デプロイに必要なパラメータの詳細を収集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。

> **注意:** このソリューションの実行中に使用した AWS サービスのコストは、お客様の負担となります。詳細については、このガイドの「コスト」セクションで、このソリューションで使用されている各 AWS サービスの料金表ウェブページを参照してください。

1. AWS マネジメントコンソールにサインインして、`workload-discovery-on-aws.template` AWS CloudFormation テンプレートを起動するボタンを選択します。

   ソリューションの起動

   あるいは、独自にカスタマイズする場合は、テンプレートをダウンロードすることもできます。

2. テンプレートは、デフォルトで米国東部 (バージニア北部) リージョンで起動されます。別の AWS リージョンでこのソリューションを起動するには、コンソールのナビゲーションバーのリージョンセレクターを使用します。

> **注意**: このソリューションでは、一部の AWS リージョンでしか利用できないサービスを使用しています。サポートされている AWS リージョンのリストについては、「利用可能なデプロイリージョン」を参照してください。

3. **スタックの作成**ページで、正しいテンプレート URL が [**Amazon S3 URL**] テキストボックスに示されていることを確認し、[**次へ**] を選択します。
4. **スタックの詳細を指定**ページで、このソリューションのスタックに名前を割り当てます。名前に使用する文字の制限に関する詳細については、*AWS Identity and Access Management* ユーザーガイドの「IAM および STS クォータ」を参照してください。
5. **パラメータ**で、このソリューションのテンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

パラメータ	デフォルト	説明
AdminUserEmailAddress	<入力が必須>	最初に作成するユーザーの電子メールアドレス。一時的な認証情報は、この E メールアドレスに送信されます。

パラメータ	デフォルト	説明
AlreadyHaveConfigSetup	`No`	デプロイ用アカウントに AWS Config が既に設定されているかどうかの確認。詳細については、「前提条件」を参照してください。
AthenaWorkgroup	`primary`	コスト機能が有効な場合に、Amazon Athena クエリの発行に使用されるワークグループ。
CreateNeptuneReplica	`No`	別のアベイラビリティーゾーンで Amazon Neptune のリードレプリカを作成するかどうかを選択します。`Yes` を選択すると、回復力は向上しますが、このソリューションのコストは増加します。
CreateOpenSearchServiceRole	`Yes`	Amazon OpenSearch Service にリンクされたロールを既に持っているかどうかの確認。詳細については、「前提条件」を参照してください。
NeptuneInstanceClass	`db.r5.large`	Amazon Neptune データベースをホストするために使用するインスタンスタイプ。ここで選択する内容は、このソリューションを実行するコストに影響します。
OpensearchInstanceType	`m6g.large.search`	Amazon OpenSearch Service のデータノードに使用するインスタンスタイプ。選択したインスタンスタイプは、このソリューションの実行コストに影響します。
OpensearchMultiAz	`No`	複数のアベイラビリティーゾーンにまたがる Amazon OpenSearch Service クラスターを作成するかどうかを選択します。`Yes` を選択すると、回復力は向上しますが、このソリューションのコストは増加します。
CreateAPIGatewayCloudWatchLogsRole	`Yes`	Yes に設定すると、このソリューションはロールを作成し、既存の `APIGatewayCloudWatchLogsLogsRole` プロパティを上書きします。既存のロールセットが既にある場合は、`No` に設定してください。詳細については、「前提条件」を参照してください。

パラメータ	デフォルト	説明
OptOutOfSendingAnonymousUsageMetrics	No	AWS への基本的な使用状況メトリクスの送信を停止するかどうか選択します。

6. [**次へ**] を選択します。
7. **スタックオプションの設定**ページで、[**次へ**] を選択します。
8. **レビュー**ページで、設定を見直して確認します。テンプレートによって AWS Identity and Access Management (IAM) リソースと必要な追加機能が作成されることを承認するチェックボックスをオンにします。
9. [**スタックの作成**] を選択してスタックをデプロイします。

スタックのステータスは、AWS CloudFormation コンソールのステータス列で確認できます。約 30 分で **CREATE_COMPLETE** ステータスが表示されます。

> **注意**: 削除すると、このスタックによってすべてのリソースが削除されます。スタックが更新された場合は、設定されたユーザーが失われないように Amazon Cognito ユーザープールが保持されます。

デプロイ後のタスク設定の確認

「AWS でのワークロード検出」ソリューションが正常にデプロイされたら、次のデプロイ後の設定を確認します。

Amazon Cognito で高度なセキュリティを有効にする

Amazon Cognito の高度なセキュリティ機能を有効にするには、*Amazon Cognito* 開発者ガイドの「ユーザープールに高度なセキュリティを追加する」の手順に従ってください。

> **注意:** Amazon Cognito で高度なセキュリティを有効にするには追加料金がかかります。

Amazon Cognito ユーザーの作成

「AWS でのワークロード検出」ソリューションでは、Amazon Cognito を使用してすべてのユーザーと認証を管理します。デプロイ中にユーザーが作成され、一時的な認証情報で提供されたアドレスに E メールが送信されます。

ユーザーの追加

1. AWS Cognito コンソールにサインインします。
2. [**ユーザープールの管理**] を選択します。
3. [WD**CognitoUserPool-***<ID-string>*] を選択します。
4. ナビゲーションペインの**全般設定**で、[**ユーザーとグループ**] を選択します**。**
5. **ユーザー**タブで、[**ユーザーの作成**] を選択します。
6. **ユーザーの作成**ボックスで、すべての必須フィールドに値を入力します。

フォームのフィールド	必須?	説明
ユーザー名	Yes	「AWS でのワークロード検出」ソリューションへのログインに使用するユーザー名。
この新規ユーザーに招待を送信しますか?	はい (E メールのみ)	選択すると、仮パスワードのリマインダーとして通知が送信されます。[**E メール**] のみを選択してください。[**SMS (デフォルト)**] を選択すると、エラーメッセージが表示されますが、ユーザーは作成されます。
仮パスワード	Yes	仮パスワードを入力します。 ユーザーは、「AWS でのワークロード検出」ソリューションに初めてログインするときに、パスワードの変更を強制されます。
電話番号	No	電話番号を国際形式で入力します（例: +44）。「**電話番号を検証済みにしますか?**」ボックスが選択されていることを確認してください。

フォームのフィールド	必須?	説明
E メール	Yes	有効な E メールアドレスを入力します。「**E メールを検証済みにしますか?**」ボックスが選択されていることを確認してください。

7. [**ユーザーの作成**] を選択します。

このプロセスを繰り返して、必要な数のユーザーを作成します。

> **注意:** すべてのユーザーは、検出されたリソースに対する同じレベルのアクセス権を持ちます。機密性の高いワークロードやデータを含む AWS アカウントには、「AWS でのワークロード検出」ソリューションを個別にデプロイしてプロビジョニングすることをお勧めします。これにより、アクセスを必要とするユーザーだけにアクセスを制限できます。

「AWS でのワークロード検出」ソリューションへのログイン

このソリューションが正常にデプロイされたら、ウェブ UI で使用する Amazon CloudFront ディストリビューション の URL を決定します。

1. AWS CloudFormation コンソールにサインインします。
2. [**ネストを表示**] を選択して、デプロイを構成するネストされたスタックを表示します。設定によっては、ネストされたスタックが既に表示されている場合があります。
3. メインのワークロード検出スタックを選択します。
4. [**出力**] タブを選択し、**WebUiURL** キーに関連する**値**列の URL を選択します。
5. **サインイン**画面で、E メールで受け取ったユーザー名とパスワードを入力します。次に、次のアクションを実行します。
   a. プロンプトに従ってパスワードを変更します。
   b. E メールに送信された検証コードを使用して、アカウントを有効化します。

AWS リージョンのインポート

インポートする AWS リージョンに特定のインフラストラクチャをデプロイする必要があります。このインフラストラクチャは、***Global*** と ***Regional*** のリソースで構成されています。

Global - AWS アカウントに 1 度だけデプロイされ、インポートされた AWS リージョンごとに再利用されるリソース。

- IAM ロール (`WorkloadDiscoveryRole`)

Regional - インポートされた AWS リージョンごとにデプロイされるリソース。

- AWS Config の配信チャネル
- AWS Config 用の Amazon S3 バケット
- IAM ロール (`ConfigRole`)

このインフラストラクチャのデプロイには、次の 2 つのオプションがあります。

- AWS CloudFormation StackSets (推奨)
- AWS CloudFormation

AWS CloudFormation StackSets

次の手順では、AWS CloudFormation StackSets を使用して、AWS リージョンをインポートし、AWS CloudFormation テンプレートをデプロイする方法について説明します。

1. 「AWS Perspective」ソリューションにサインインします。URL については、「ソリューションへのログイン」を参照してください。
2. サイドナビゲーションパネルで、[**Accounts**] を選択します。
3. 画面の右上隅にある [**Import**] ボタンを選択します。
4. ラジオボタンでインポート方法を選択します。
    a. CSV ファイルを使用して、**Accounts と Regions を追加します**。
    b. フォームを使用して、**Accounts と Regions を追加します**。

CSV ファイル

次のフォーマットでインポートされた AWS リージョンを含むカンマ区切り値 (CSV) を指定します。

```
"accountId","accountName","region"
123456789012,"test-account-1",eu-west-2
123456789013,"test-account-2",eu-west-1
123456789013,"test-account-2",eu-west-2
123456789014,"test-account-3",eu-west-3
```

1. [**Upload a CSV**] を選択します。
2. CSV ファイルを見つけて開きます。
3. **Regions** テーブルを確認して、[**Import**] を選択します。
4. モーダルダイアログで、Global のリソーステンプレートと Regional のリソーステンプレートをダウンロードします。
5. AWS CloudFormation テンプレートを関連する AWS アカウントにデプロイします (次のセクションを参照)。
6. インストールしたら、両方のチェックボックスを選択してインストールが完了したことを確認し、[**Import**] ボタンを選択します。

フォーム

次のフォームを使用して、インポートする AWS リージョンを指定します。

1. **Account ID**: 12 桁のアカウント ID を入力するか、既存のアカウント ID を選択します。
2. **Account name**: アカウント名を入力するか、既存のアカウント ID を選択したときに事前に入力した値を使用します。
3. **Regions**: インポートする AWS リージョンを選択します。
4. [**Add**] を選択して、次の **Regions** テーブルに AWS リージョンを入力します。
5. **Regions** テーブルを確認して、[**Import**] を選択します。
6. モーダルダイアログで、Global のリソーステンプレートと Regional のリソーステンプレートをダウンロードします。
7. AWS CloudFormation テンプレートを関連する AWS アカウントにデプロイします (次のセクションを参照)。

8. インストールしたら、両方のチェックボックスを選択してインストールが完了したことを確認し、[**Import**] ボタンを選択します。

AWS CloudFormation テンプレートをデプロイする

Global リソースは、AWS アカウントごとに 1 度デプロイする必要があります。「AWS でのワークロード検出」ソリューションにインポート済みの AWS リージョンを含む AWS アカウントから AWS リージョンをインポートする場合は、このテンプレートをデプロイしないでください。AWS リージョンが既にインポートされている場合は、「AWS CloudFormation を使用してスタックをデプロイし、Regional リソースをプロビジョニング」に進んでください。

AWS CloudFormation StackSet を使用して、複数のアカウント間で Global リソースをプロビジョニング

> **重要:** まず、StackSets オペレーションの前提条件を完了してから、ターゲットの AWS アカウントで StackSet を有効にしてください。

1. 管理者アカウントで、AWS CloudFormation コンソールにサインインします。
2. 左側のナビゲーションパネルから [**StackSets**] を選択します。
3. [**StackSet の作成**] を選択します。
4. **テンプレートの選択**ページの**テンプレートの指定**で、[**テンプレートファイルのアップロード**] を選択し、前にダウンロードした `global-resources.template` ファイルを選択して、[**次へ**] を選択します。
5. **StackSet の詳細を指定**ページで、StackSet に名前を割り当てます。名前に使用する文字の制限に関する詳細については、*AWS Identity and Access Management* ユーザーガイドの「IAM および STS クォータ」を参照してください。
6. **パラメータ**で、このソリューションのテンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

フィールド名	デフォルト	説明
AccountId	デプロイ用アカウント ID	デプロイ用の AWS アカウントとして最初に作られたアカウント ID。デフォルトのままにしてください。

7. [**次へ**] を選択します。
8. AWS Organization で StackSets を使用している場合:
   [**サービスマネージド型のアクセス許可**] または [**セルフマネージド型のアクセス許可**] を選択します。詳細については、「AWS Organization での StackSet の使用」を参照してください。

   AWS Organizations を使用していない場合:
   StackSets の前提条件の手順に従うときに使用する IAM 実行ロール名を入力します。詳しくは、「セルフマネージド型のアクセス許可を付与する」を参照してください。
9. [**次へ**] を選択します。
10. **StackSet にスタックを追加**の**アカウント番号**ボックスで、アカウントロールをデプロイするためのアカウント ID を入力します。
11. **リージョンの指定**で、スタックをインストールする **AWS リージョン**を 1 つ選択します。
12. **デプロイオプション**で [**並行**] を選択し、[**次へ**] を選択します。
13. AWS CloudFormation がカスタム名で IAM リソースを作成する可能性があることを承認するチェックボックスをオンにします。[**送信**] を選択します。

AWS CloudFormation StackSets を使用して、Regional リソースをプロビジョニング

> **重要:** まず、スタックセットオペレーションの前提条件を完了してから、ターゲットの AWS アカウントで StackSet を有効にしてください。
>
> AWS Config がインストールされている AWS リージョンとインストールされていない AWS リージョンがある場合は、それぞれに対して StackSet オペレーションを実行する必要があります。

1. 管理者アカウントで、AWS CloudFormation コンソールにサインインします。
2. 左側のナビゲーションパネルから [**StackSets**] を選択します。
3. [**StackSet の作成**] を選択します。
4. **テンプレートの選択**ページの**テンプレートの指定**で、[**テンプレートファイルのアップロード**] を選択し、前にダウンロードした `regional-resources.template` ファイルを選択して、[**次へ**] を選択します。
5. **StackSet の詳細を指定**ページで、StackSet に名前を割り当てます。名前に使用する文字の制限に関する詳細については、*AWS Identity and Access Management* ユーザーガイドの「IAM および STS クォータ」を参照してください。
6. **パラメータ**で、このソリューションのテンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

フィールド名	デフォルト	説明
AccountId	デプロイ用アカウント ID	デプロイ用のアカウントとして最初に作られたアカウント ID。デフォルトのままにしてください。
AggregationRegion	デプロイした AWS リージョン	最初にデプロイされた AWS リージョン。デフォルトのままにしてください。
AlreadyHaveConfigSetup	`No`	AWS リージョンに AWS Config が既に有効になっているか確認します。AWS Config がこの AWS リージョンで既に有効になっている場合は、`Yes` に設定します。

7. [**次へ**] を選択します。
8. AWS Organization で StackSets を使用している場合:
   [**サービスマネージド型のアクセス許可**] または [**セルフマネージド型のアクセス許可**] を選択します。詳細については、「AWS Organization での StackSet の使用」を参照してください。

   AWS Organizations を使用していない場合:
   StackSets の前提条件の手順に従うときに使用する IAM 実行ロール名を入力します。詳しくは、「セルフマネージド型のアクセス許可を付与する」を参照してください。

9. [**次へ**] を選択します。
10. **StackSet にスタックを追加**の**アカウント番号**ボックスに、アカウントロールをデプロイするためのアカウント ID を入力します。
11. **リージョンの指定**で、スタックをインストールする **AWS リージョン**を 1 つ選択します。手順 6 で入力したすべての AWS アカウントにスタックがインストールされます。
12. **デプロイオプション**で [**並行**] を選択し、[**次へ**] を選択します。
13. AWS CloudFormation がカスタム名で IAM リソースを作成する可能性があることを承認するチェックボックスをオンにします。[**送信**] を選択します。

AWS CloudFormation を使用してスタックをデプロイし、Global リソースをプロビジョニング

Global リソースは、AWS アカウントごとに 1 度デプロイする必要があります。「AWS でのワークロード検出」ソリューションにインポート済みの AWS リージョンを含む AWS アカウントから AWS リージョンをインポートする場合は、このテンプレートをデプロイしないでください。

1. AWS CloudFormation コンソールにサインインします。
2. [**スタックの作成**] を選択し、[**新しいリソースを使用 (標準)**] を選択します。
3. **スタックの作成**ページの**テンプレートの指定**セクションで、[**テンプレートファイルのアップロード**] を選択します。
4. [**ファイルの選択**] を選択してから、ダウンロードした `global-resources.template` ファイルを選択して、[**次へ**] を選択します。
5. **スタックの詳細を指定**ページで、このソリューションのスタックに名前を割り当てます。名前に使用する文字の制限に関する詳細については、*AWS Identity and Access Management* ユーザーガイドの「IAM および STS クォータ」を参照してください。
6. **パラメータ**で、このソリューションのテンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

フィールド名	デフォルト	説明
Stack Name	`workload-discovery`	この AWS CloudFormation スタックの名前。
AccountId	デプロイ用アカウント ID	デプロイ用の AWS アカウントとして最初に作られたアカウント ID。デフォルトのままにしてください。

7. [**次へ**] を選択します。
8. AWS CloudFormation がカスタム名で IAM リソースを作成する可能性があることを承認するチェックボックスをオンにします。
9. [**スタックを作成**] を選択します。

新しい AWS リージョンでは 15 分間隔で実行される検出プロセスでスキャンされます。(例: 15:00、15:15、15:30、15:45)

「AWS でのワークロード検出」ソリューションのウェブ UI に移動して、サイドナビゲーションペインで次の検出までの推定時間を検索します。

想定されるリソースがウェブ UI に表示されない場合は、「AWS リージョンが正しくインポートされたことを確認」を参照してください。

AWS CloudFormation を使用してスタックをデプロイし、Regional リソースをプロビジョニング

1. AWS CloudFormation コンソールにサインインします。
2. [**スタックの作成**] を選択し、[**新しいリソースを使用 (標準)**] を選択します。
3. **スタックの作成**ページの**テンプレートの指定**セクションで、[**テンプレートファイルのアップロード**] を選択します。
4. [**ファイルの選択**] を選択し、ダウンロードした `regional-resources.template` ファイルを選択して、[**次へ**] を選択します。

5. **スタックの詳細を指定**ページで、このソリューションのスタックに名前を割り当てます。名前に使用する文字の制限に関する詳細については、*AWS Identity and Access Management* ユーザーガイドの「IAM および STS クォータ」を参照してください。
6. **パラメータ**で、このソリューションのテンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

フィールド名	デフォルト	説明
Stack Name	*workload-discovery*	この AWS CloudFormation スタックの名前。
AccountId	デプロイ用アカウント ID	デプロイ用の AWS アカウントとして最初に作られたアカウント ID。デフォルトのままにしてください。
AggregationRegion	デプロイした AWS リージョン	最初にデプロイされた AWS リージョン。デフォルトのままにしてください。
AlreadyHaveConfigSetup	`No`	AWS リージョンに AWS Config が既に有効になっているか確認します。AWS Config がこの AWS リージョンで既に有効になっている場合は、`Yes` に設定します。

7. [**次へ**] を選択します。
8. AWS CloudFormation がカスタム名で IAM リソースを作成する可能性があることを承認するチェックボックスをオンにします。
9. [**スタックを作成**] を選択します。

   新しい AWS リージョンは、15 分間隔で実行される検出プロセスでスキャンされます。(例: 15:00、15:15、15:30、15:45)

   「AWS でのワークロード検出」ソリューションのウェブ UI に移動して、サイドナビゲーションペインで次の検出までの推定時間を検索します。

   想定されるリソースがウェブ UI に表示されない場合は、「AWS リージョンが正しくインポートされたことを確認」を参照してください。

AWS リージョンが正しくインポートされたことを確認

1. サインインします (既にページを表示済みの場合は更新してください)。URL については、「ソリューションへのログイン」を参照してください。
2. 左側のナビゲーションパネルの **Settings** で、[**Imported Regions**] を選択します。

   AWS リージョン、アカウント名、アカウント ID がテーブルに表示されます。**Last Scanned** 列には、その AWS リージョンでリソースを最後に検出した時刻が表示されます。

コスト機能の設定

コスト機能を使用するには、コストと使用状況レポート (CUR) を手動で設定する必要があります。

1. スケジュールされたコストと使用状況レポートを設定します。
2. Amazon S3 のレプリケーションを設定します (CUR がデプロイ用アカウントと別のアカウントにある場合)。

デプロイ用アカウントで *AWS* のコストと使用状況レポートを作成する

1. コストデータを収集する AWS アカウントの Billing コンソールにサインインします。
2. 左側のナビゲーションペインの **Billing** カテゴリで、[**Cost & Usage Reports**] を選択します。
3. [**レポートの作成**] を選択します。
4. **レポート名**には `workload-discovery-cost-and-usage-<your-workload-discovery-deployment-account-ID>` を使用します。

   > **注意**: CUR のクエリを容易にするために少量のインフラストラクチャがデプロイされるため、この命名規則に従う必要があります。

5. [**リソース ID のインクルード**] チェックボックスをオンにします。

> **注意**: コストデータを表示するには、[**リソース ID のインクルード**] ボックスを選択する必要があります。この ID は、「AWS でのワークロード検出」ソリューションで検出されたリソースと一致させるのに必要です。

6. [**次へ**] を選択します。
7. 配信オプションページで、[**設定**] を選択します。
8. `<stack-name>-s3buc-costandusagereportbucket-<ID-string>` Amazon S3 バケットを選択して、CUR を保存します。[**次へ**] を選択します。
9. ポリシーを確認し、確認ボックスをオンにして、[**保存**] を選択します。
10. **レポートパスのプレフィックス**を `workload-discovery` に設定します。
11. 時間単位には [**日別**] を選択します。
12. **レポートデータ統合の有効化**で、[**Amazon Athena**] を選択します。
13. [**次へ**] を選択します。
14. [**確認して完了**] を選択します。

レポートが正しくセットアップされていることを確認するには、Amazon S3 バケットでテストファイルを確認します。

> **注意**: レポートが Amazon S3 バケットにアップロードされるまでに、最大で 24 時間かかる場合があります。

他のアカウントで *AWS* のコストと使用状況レポートを作成する

1. コストデータを収集する AWS アカウントの Billing コンソール にサインインします。
2. 左側のナビゲーションペインの **Billing** カテゴリで、[**Cost & Usage Reports**] を選択します。
3. [**レポートの作成**] を選択します。
4. **レポート名**には `workload-discovery-cost-and-usage-<your-external-account-ID>` を使用します。

> **注意**: CUR のクエリを容易にするために少量のインフラストラクチャがデプロイされるため、この命名規則に従う必要があります。

5. [**リソース ID のインクルード**] チェックボックスをオンにします。

> **注意**: コストデータを表示するには、[**リソース ID のインクルード**] ボックスを選択する必要があります。この ID は、「AWS でのワークロード検出」ソリューションで検出されたリソースと一致させるために必要です。

6. [**次へ**] を選択します。
7. 配信オプションページで、[**設定**] を選択します。
8. CUR を保存する新しい Amazon S3 バケットを作成します。
9. ポリシーを確認し、確認ボックスをオンにして、[**保存**] を選択します。
10. **レポートパスのプレフィックス**を `workload-discovery` に設定します。
11. 時間単位には [**日別**] を選択します。
12. **レポートデータ統合の有効化**で、[**Amazon Athena**] を選択します。
13. [**次へ**] を選択します。
14. [**確認して完了**] を選択します。
    レポートが正しくセットアップされていることを確認するには、Amazon S3 バケットでテストファイルを確認します。

> **注意**: レポートが Amazon S3 バケットにアップロードされるまでに、最大で 24 時間かかる場合があります。

次に、デプロイ用アカウントへのレプリケーションを設定します。

レプリケーションを設定する

デプロイ時に作成された Amazon S3 バケットへのレプリケーションを設定します。Amazon S3 バケットは、*`<stack-name>`*`-s3buc-costandusagereportbucket-`*`<ID-string>`* の形式に従います。これにより、Amazon Athena を使用してクエリを実行できるようになります。

1. レプリケートする必要がある作成済みの CUR を使用して、Amazon S3 コンソールで AWS アカウントにサインインします。
2. AWS のコストと使用状況レポートを設定するときに作成された Amazon S3 バケットを選択します。(「デプロイ用アカウントで AWS のコストと使用状況レポートを作成する」の手順 8)
3. [**管理**] タブを選択します。
4. **レプリケーションルール**で、[**レプリケーションルールを作成**] を選択します。
5. **レプリケーションルールの設定**の [**レプリケーションルール名**] ボックスに、分かりやすいルール ID を入力します。
6. **ソースバケット**で [**バケット内のすべてのオブジェクトに適用**] を選択して、ルールスコープを設定します。
7. **送信先**で、次の項目を設定します。
   a. [**別のアカウントのバケットを指定する**] を選択します。
   b. アカウント ID を入力します。
   c. 「AWS でのワークロード検出」ソリューションのデプロイ時に作成された Amazon S3 の**バケット名**を入力します。これは、論理 ID (`CostAndUsageReportBucket`) と「AWS でのワークロード検出」ソリューションを最初にデプロイしたときに指定したスタック名を使用して、「デプロイリソースの検索」の手順に従うことで見つけることができます。
   d. [**オブジェクト所有者を送信先バケット所有者に変更**] チェックボックスをオンにします。
8. **IAM ロール**で、[**新しいロールの作成**] を選択します。

> **注意**: レプリケーションロールが既に存在している場合があります。これを選択して、必要な Amazon S3 のレプリケーションロールのアクションがあることを確認できます。

9. [**保存**] を選択します。
10. CUR がインストールされている AWS マネジメントコンソールにログインし、**Amazon S3** のサービスページに移動して、`CostAndUsageReportBucket` Amazon S3 バケットを選択します。詳細については、「デプロイリソースの検索」を参照してください。

11. [**管理**] タブを選択します。
12. **レプリケーションルール**の**アクション**ドロップダウンメニューから、[**レプリケートされたオブジェクトの受信**] を選択します。
13. [**ソースバケットアカウント設定**] で、次を設定します。
    a. ソースバケットのアカウント ID を入力します。
    b. [**ポリシーの生成**] を選択します。
    c. **ポリシー**で、 [**バケットポリシーの表示**] を選択します。
    d. [**オブジェクト所有者を送信先バケット所有者に変更するアクセス許可を含める**] を選択します。
    e. [**設定の適用**] を選択します。これにより、オブジェクトをコピーするアクセス権が付与されます。Amazon S3 バケットのポリシー例については、「Amazon S3 バケットポリシー」を参照してください。

> **注意**: 複数の AWS アカウントから CUR をレプリケートする場合は、送信先バケット (「AWS でのワークロード検出」ソリューションの AWS アカウント内) のバケットポリシーに、AWS アカウントごとに使用している各 IAM ロールの ARN があることを確認する必要があります。詳細については、「Amazon S3 バケットポリシー」を参照してください。

レポートが AWS アカウントにある場合は、コストデータは境界ボックスと個々のリソースに表示されます。

図 *8:* 予想コストデータを含む境界ボックスの例

Amazon S3 バケットのライフサイクルポリシーの編集

デプロイ時に、次の 2 つの Amazon S3 バケットにライフサイクルポリシーを設定します。

- `CostAndUsageReportBucket`
- `AccessLogsBucket`

> **重要:** このライフサイクルポリシーは、90 日後にこれらの Amazon S3 バケットからデータを削除します。ライフサイクルを編集して、既存のポリシーに合わせることができます。

その他のリソース

AWS のサービス

- Amazon Athena
- Amazon CloudFront
- Amazon CloudWatch
- Amazon Cognito
- Amazon DynamoDB
- Amazon Elastic Container Registry
- Amazon Elastic Container Service
- Amazon OpenSearch Service
- Amazon Neptune
- Amazon Simple Notification Service
- Amazon Simple Storage Service
- Amazon Virtual Private Cloud
- AWS AppSync
- AWS CloudFormation
- AWS CodeBuild
- AWS Config
- AWS Fargate
- AWS Identity and Access Management (IAM)
- AWS Lambda
- AWS SDK for JavaScript
- AWS Systems Manager

AWS API

- describeVpcEndpoints
- describeSpotFleetRequests
- describeSpotInstanceRequests
- describeDBClusters
- getAccountAuthorizationDetails
- describeInstances
- describeLoadBalancers
- describeListeners
- describeTargetGroups
- describeTargetHealth
- getFunction
- GetFunctionConfiguration
- describeTaskDefinition
- listTasks
- describeTasks
- listServices
- describeServices
- listClusters
- describeClusters
- listContainerInstances
- describeContainerInstances
- getRestApis
- getResources
- getIntegration

スタックの更新

「AWS でのワークロード検出」ソリューションの v1.x.x から v2.0.0 へのアップデートはサポートされていません。v2.0.0 をインストールする前に、「AWS でのワークロード検出」ソリューションの v1.x.x をアンインストールすることをお勧めします。

v2.x.x からアップデートする場合は、次の手順に従います。

1. このソリューションの AWS CloudFormation テンプレートをダウンロードします。
2. AWS CloudFormation コンソールにサインインします。
3. デプロイ時に指定した名前のスタックを選択し、[**更新**] を選択します。
4. **スタックの更新**ページで、[**既存テンプレートを置き換える**] を選択し、[**テンプレートファイルのアップロード**] を選択して、手順 1 でダウンロードしたファイルをアップロードします。
5. [**次へ**] を選択します。

6. **スタックの詳細を指定**ページの**パラメータ**で、パラメータを確認し、必要に応じて変更します。
7. [**次へ**] を選択します。
8. **スタックオプションの設定**ページで、[**次へ**] を選択します。
9. **レビュー**ページで、設定を見直して確認します。テンプレートによって AWS Identity and Access Management (IAM) リソースと必要な追加機能が作成されることを承認するチェックボックスをオンにします。
10. [**スタックの更新**] を選択して、スタックをデプロイします。

デプロイリソースの検索

次の手順に従って、AWS アカウントにデプロイしたリソースを検索します。

1. AWS CloudFormation コンソールにサインインします。
2. デプロイした AWS リージョンを選択します。

このアカウントの使用状況によっては、ワークロードごとに複数のスタックが含まれる場合があります。デプロイ時に指定した名前のメインスタックと、その下に複数のネストされたスタックがあります。

3. 各スタックを選択して、そのテンプレートを使用してデプロイされたリソースにアクセスします。
4. [**リソース**] タブを選択し、関連するリソースの [**物理 ID**] リンクを選択して、それぞれのサービスコンソールでリソースを表示します。

リソースの**論理 ID** がわかっている場合は、検索を使用することもできます。

サポートされているリソース

このソリューションでは、ここにリストされているように、AWS Config がサポートするすべてのリソースをサポートしています。次の表には、AWS Config でサポートされていない、「AWS でのワークロード検出」ソリューションが検出するサポート対象のリソースが含まれています。詳細については、対応する AWS ドキュメントのリストを参照してください。

リソースタイプ	ソース	説明
AWS::ApiGateway::Authorizer	SDK	getAuthorizers
AWS::ApiGateway::Resource	SDK	getResource
AWS::ApiGateway::Method	SDK	getMethod
AWS::Cognito::UserPool	SDK	describeUserPool
AWS::ECS::Task	SDK	describe-tasks
AWS::EKS::Nodegroup	SDK	describeNodegroup
AWS::IAM::AWSManagedPolicy	SDK	getAccountAuthorizationDetails
AWS::ElasticLoadBalancingV2::TargetGroup	SDK	describeTargetGroups
AWS::EC2::Spot	SDK	describeSpotInstanceRequests
AWS::EC2::SpotFleet	SDK	describeSpotFleetRequests

AWS Organization での StackSet の使用

AWS Organization で StackSets を使用している場合は、組織が StackSet の IAM アクセス権限を管理する方法に応じて、サービスマネージド型のアクセス許可またはセルフマネージド型のアクセス許可のいずれかを選択します。組織への StackSet のデプロイの詳細については、AWS ブログの「新機能: AWS CloudFormation StackSets が AWS Organization のマルチアカウントで利用可能に」を参照してください。

トラブルシューティング

「AWS でのワークロード検出」ソリューションのスタックが AWS アカウントにデプロイされたときに失敗した場合、次を確認してください。

- `ConfigAggregtor` スタックにはエラーはありません。配信チャネルのエラーがある場合は、AWS Config がすでに有効になっているということです。`AlreadyHaveConfigSetup` パラメータを `Yes` に設定してください。
- `CreateOpensearchServiceRole` が `Yes` に設定されている場合は、`AWSServiceRoleForAmazonElasticsearchService` IAM ロールがアカウントにまだ存在していないことを確認してください。

> **注意:** トラブルシューティングを容易にするには、AWS CloudFormation テンプレートで失敗時のロールバック機能を無効にすることをお勧めします。また、「AWS でのワークロード検出」ソリューションのドキュメントには、トラブルシューティングに関するその他のヘルプもあります。

Amazon S3 のレプリケーションロールのアクション

レプリケーションの実行に使用される IAM ロールには、次のアクションが必要です。

s3:ReplicateObject
s3:ReplicateDelete
s3:ReplicateTags
s3:ObjectOwnerOverrideToBucketOwner
s3:ListBucket
s3:GetReplicationConfiguration
s3:GetObjectVersionForReplication

s3:GetObjectVersionAcl
s3:GetObjectVersionTagging
s3:GetObjectRetention
s3:GetObjectLegalHold

ロールにレプリケーションロールのアクションがあることを確認するには、次の手順を実行します。

1. Amazon S3 レプリケーションウィザードで、ロール名の名前をコピーします。
2. レプリケーションを設定する AWS アカウント内の IAM コンソールに移動します。
3. ロールの名前を**検索**ボックスに貼り付けます。
4. リストから最上位の項目を選択します。これが使用する IAM ロールです。
5. **許可ポリシー**内の**ポリシー**を展開します。
6. 上表に詳述されているアクションが定義されていることを確認します。

Amazon S3 バケットポリシー

次は、Cost and Usage Reports (CUR) をバケットにアップロードすることを許可し、外部アカウントがオブジェクトをバケットにレプリケートすることを許可する権限を持つ Amazon S3 バケットポリシーの例です。レプリケーションを実行するためのアクセス権限を付与するには、外部の各 AWS アカウントの IAM ロールをこのポリシーに追加する必要があります。

```
{
   "Version":"2012-10-17",
   "Id":"",
   "Statement":[
      {
         "Sid":"Set permissions for objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn-of-role-selected-in-replication-setup-in-
source-account"
         },
         "Action":["s3:ReplicateObject", "s3:ReplicateDelete"],
"s3:ObjectOwnerOverrideToBucketOwner",
         "Resource":"arn:aws:s3:::destination-bucket-name/*"
```

```
        },
        {
            "Sid":"Set permissions on bucket",
            "Effect":"Allow",
            "Principal":{
               "AWS":"arn-of-role-selected-in-replication-setup-in-
source-account"
            },
            "Action":["s3:GetBucketVersioning",
"s3:PutBucketVersioning"],
            "Resource":"arn:aws:s3:::destination-bucket-name "
        },
        {
              "Sid": "Stmt1335892150622",
              "Effect": "Allow",
              "Principal": {
                  "Service": "billingreports.amazonaws.com"
              },
              "Action": [
                  "s3:GetBucketAcl",
                  "s3:GetBucketPolicy"
              ],
              "Resource": "arn:aws:s3:::destination-bucket-name"
          },
          {
              "Sid": "Stmt1335892526596",
              "Effect": "Allow",
              "Principal": {
                  "Service": "billingreports.amazonaws.com"
              },
              "Action": "s3:PutObject",
              "Resource": "arn:aws:s3:::destination-bucket-name/*"
          }
    ]
}
```

ソリューションのアンインストール

このソリューションをアンインストールするには、AWS マネジメントコンソールまたは AWS コマンドラインインターフェイス (AWS CLI) を使用します。まず、Amazon ECS クラスターから実行中のすべてのタスクを停止します。そうしないと、スタックの削除が失敗する可能性があります。

AWS マネジメントコンソールの使用

1. AWS CloudFormation コンソールにサインインします。
2. デプロイ時に指定した名前のスタックを選択します。
3. [**スタックの削除**] を選択します。

AWS Command Line Interface の使用

ご使用の環境で AWS Command Line Interface (AWS CLI) が使用できるかどうかを確認します。インストール手順については、*AWS CLI* ユーザーガイドの「AWS Command Line Interface とはどのようなものですか」を参照してください。

AWS CLI が使用可能になったことを確認したら、次のコマンドを実行します。

```
$ aws cloudformation delete-stack --stack-name <customer-defined-stack-name>
```

運用メトリクスの収集

このソリューションには、匿名の運用メトリクスを AWS に送信するオプションが含まれています。当社はこのデータを使用して、お客様がこのソリューション、関連サービスおよび製品をどのように使用しているかをよりよく理解し、提供するサービスや製品の改善に役立てます。有効にすると、次の情報が収集され、AWS に送信されます。

- **Solution ID:** SO0075 SO0075a SO0075b SO0075c
- **Unique ID (UUID):** デプロイごとにランダムに生成された一意の識別子
- **Timestamp:** データ収集タイムスタンプ
- **Login Attempts:** ログインごとに増分され、匿名で送信されます。

- **Instance Data:** 各 AWS リージョンで Amazon EC2 のスケジューラによって管理されるインスタンスの状態とタイプの数

  データのサンプル:

```
Running: {t2.micro: 2}, {m3.large:2}
Stopped: {t2.large: 1}, {m3.xlarge:3}
```

AWS は、このアンケートを通じて収集されたデータを所有します。データ収集には、AWS プライバシーポリシーが適用されます。この機能を無効にするには、**OptOutOfSendingAnonymousUsageMetrics** を `Yes` に設定したこのソリューションのテンプレートをデプロイしてください。

ソースコード

「AWS でのワークロード検出」ソリューションの GitHub リポジトリにアクセスして、このソリューションのテンプレートとスクリプトをダウンロードし、カスタマイズを他のユーザーと共有できます。

改訂

日付	変更
2020 年 9 月	初回リリース
2020 年 9 月	リリースバージョン 1.0.1: バグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2021 年 8 月	リリースバージョン 1.1.0: 新機能とバグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2021 年 10 月	リリースバージョン 1.1.1: 新機能とバグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2021 年 11 月	リリースバージョン 1.1.2: バグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2022 年 2 月	リリースバージョン 1.1.3: バグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。

日付	変更
2022 年 4 月	リリースバージョン 1.1.4: バグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2022 年 10 月	リリースバージョン 2.0.0: 新機能とバグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2022 年 11 月	リリースバージョン 2.0.1: バグ修正。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。

注意

お客様は、この文書に記載されている情報を独自に評価する責任を負うものとします。このドキュメントは、(a) 情報提供のみを目的としており、(b) AWS の現行製品とプラクティスを表したものであり、予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤー、またはライセンサーからの契約義務や確約を意味するものではありません。AWS の製品やサービスは、明示または暗示を問わず、いかなる保証、表明、条件を伴うことなく「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で行われるいかなる契約の一部でもなく、そのような契約の内容を変更するものでもありません。

「AWS でのワークロード検出」ソリューションは、Apache Software Foundation で閲覧可能な Apache ライセンスバージョン 2.0 の条項に基づいてライセンスされます。