実装ガイド

AWS でのワークロード検出

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS でのワークロード検出: 実装ガイド

Copyright © 2024 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 が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

Table of Contents

ソリューションの概要 1
機能とメリット 2
ユースケース 3
概念と定義 4
アーキテクチャの概要 5
アーキテクチャ図 5
AWS Well-Architected の設計に関する考慮事項 7
オペレーショナルエクセレンス 7
セキュリティ 7
信頼性 8
パフォーマンス効率 8
コスト最適化 9
持続可能性 9
アーキテクチャの詳細 10
[Authentication mechanism] (認証メカニズム) 10
サポート リソース 10
AWS でのワークロード検出でのアーキテクチャ図の管理 10
ウェブ UI とストレージ管理 10
データコンポーネント 11
イメージデプロイコンポーネント 13
検出コンポーネント 13
コストコンポーネント 14
このソリューションで使用している AWS のサービス 15
デプロイを計画する 18
サポートしている AWS リージョン 18
コスト 19
サンプルコスト表 19
セキュリティ 21
リソースアクセス 21
ネットワークアクセス 22
アプリケーションの設定 23
クォータ 23
このソリューション内の AWS サービスのクォータ 23
AWS CloudFormation のクォータ 24

AWS Lambda のクォータ ... 24
Amazon VPC クォータ ... 24
デプロイするアカウントを選択する ... 25
ソリューションをデプロイする ... 26
デプロイプロセスの概要 ... 26
前提条件 ... 26
デプロイパラメータの詳細を収集する ... 26
AWS CloudFormation テンプレート ... 29
スタックを起動する ... 30
デプロイ後の設定作業 ... 39
Amazon Cognito で高度なセキュリティを有効にする ... 39
Amazon Cognito ユーザーの作成 ... 39
追加のユーザーを作成するには: ... 39
AWS でのワークロード検出にログインする ... 41
リージョンをインポートする ... 41
リージョンをインポートする ... 42
AWS CloudFormation テンプレートをデプロイする ... 43
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 ... 44
CloudFormation StackSets を使用して、Regional リソースをプロビジョニング ... 45
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 ... 47
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 ... 48
リージョンが正しくインポートされたことを確認する ... 49
コスト機能の設定 ... 50
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 ... 50
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 ... 51
レプリケーションの設定 ... 52
S3 バケットのライフサイクルポリシーの編集 ... 54
ソリューションのモニタリング ... 55
myApplications ... 55
CloudWatch AppInsights ... 55
ソリューションを更新する ... 57
トラブルシューティング ... 58
既知の問題解決 ... 58
配信チャネル設定のエラー ... 58

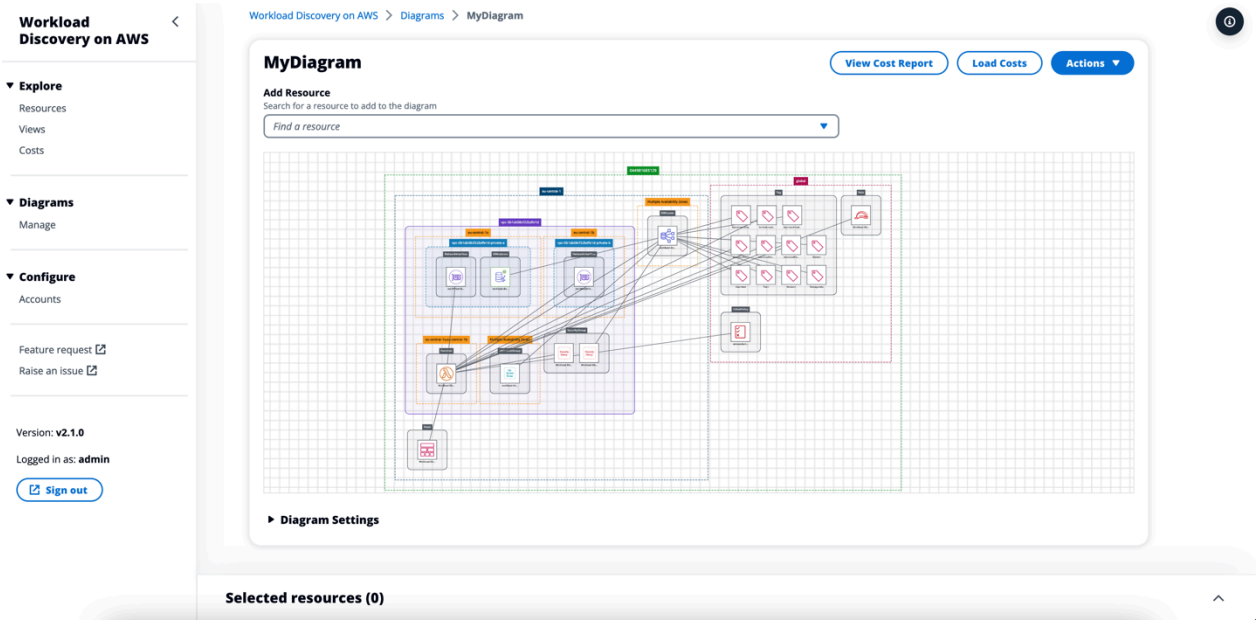
既存の VPC にデプロイすると、サーチリゾルバースタックのデプロイがタイムアウトする ... 59
アカウントのインポート後にリソースが検出されない ... 59
特定のアカウントでは AWS Config 以外のリソースのみが検出される ... 60
AWS サポートに問い合わせる ... 61
ケースを作成する ... 61
どのようなサポートをご希望ですか? ... 61
追加情報 ... 62
ケースの迅速な解決にご協力ください ... 62
今すぐ解決またはお問い合わせ ... 62
ソリューションをアンインストールする ... 63
AWS マネジメントコンソールの使用 ... 63
AWS コマンドラインインターフェイスの使用 ... 63
開発者ガイド ... 64
ソースコード ... 64
デプロイリソースの検索 ... 64
サポート リソース ... 64
AWS Organizations のアカウント検出モード ... 65
Amazon S3 レプリケーションロールのアクション ... 66
S3 バケットポリシー ... 67
AWS API ... 68
API Gateway ... 68
Cognito ... 69
構成 ... 69
DynamoDB Streams ... 69
Amazon EC2 ... 69
Amazon Elastic Load Balancer ... 69
Amazon Elastic Kubernetes Service ... 69
IAM ... 70
Lambda ... 70
OpenSearch Service ... 70
組織 ... 70
Amazon Simple Notification Service ... 70
Amazon Security Token Service ... 70
参照資料 ... 71
匿名化されたデータの収集 ... 71

寄稿者 72
改訂 73
注意 74

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生成する可視化するツールをデプロイする。

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は、AWS 上のワークロードをアーキテクチャ図として自動的に生成する可視化ツールです。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、アカウントとリージョン全体の 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。



AWS でのワークロード検出ソリューションが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を使用してこのソリューションをデプロイするために必要な 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。

AWS でのワークロード検出ソリューションをそれぞれの環境に実装する対象者には、ソリューションアーキテクト、ビジネスの意思決定者、DevOps エンジニア、データサイエンティスト、クラウドプロフェッショナルが含まれます。

このナビゲーションテーブルを使用すると、次の質問に対する回答をすばやく見つけることができます。

質問内容	参照先
このソリューションの実行に必要なコストを確認する。 米国東部 (バージニア北部) リージョンでこのソリューションを実行するための推定コストは、1 か月あたり 425.19 USD です。	コスト
このソリューションのセキュリティ上の考慮事項を理解する。	セキュリティ
このソリューションのクォータを計画する方法を確認する。	クォータ
どの AWS リージョンでこのソリューションをサポートしているのかを確認する。	サポートしている AWS リージョン
このソリューションに含まれている AWS CloudFormation テンプレートを表示またはダウンロードして、このソリューションのインフラストラクチャリソース (スタック) を自動的にデプロイする。	AWS CloudFormation テンプレート
ソースコードにアクセスする。	GitHub リポジトリ

機能とメリット

AWS でのワークロード検出には次の機能があります。

ほぼリアルタイムのデータを使用したアーキテクチャ図の作成

AWS でのワークロード検出は、15 分ごとにアカウントをスキャンして、作成した図がワークロードを正確かつ最新に表していることを確認します。

複数のアカウントとリージョンのリソースを 1 か所で表示

このソリューションは、一元化されたグラフデータベースに AWS アカウントとリージョン全体の AWS リソースのインベントリを管理するため、複数のアカウントとリージョン、およびそれらの相互関係を単一の UI で調べることができます。

AWS Organizations の統合

AWS Organizations を使用してソリューションをデプロイすると、AWS でのワークロード検出は組織内のサポートされているすべてのリソースを自動的に検出します。この構成では、アカウント固有の CloudFormation テンプレートのデプロイを直接管理して、これらのアカウントを検出できるようにする必要はありません。

ワークロード全体のコストデータを照合

コスト機能を有効にすると、アカウント内のリソースをコストで検索し、見つかったリソースを図に追加できます。既存の図にコストデータを追加することもできます。

diagrams.net (旧 draw.io) にエクスポート

AWS でのワークロード検出では、図をエクスポートして、このサードパーティー製の描画ソフトウェアを使用してさらに注釈を付けることができます。

AWS Service Catalog AppRegistry と AWS Systems Manager の機能である Application Manager との統合

このソリューションには、CloudFormation テンプレートとその基盤となるリソースを、Service Catalog AppRegistry と Application Manager の両方にアプリケーションとして登録するための Service Catalog AppRegistry リソースが含まれています。この統合により、ソリューションのリソースを一元管理し、アプリケーションの検索、レポート、管理アクションを有効にできます。

ユースケース

設計とセキュリティのレビュー

このソリューションを使用してアーキテクチャ図を作成し、ワークロードの実装が提案された設計と一致することを検証します。

既存ワークロードの調査とドキュメント化

アーキテクチャ図を作成して、ドキュメントがほとんど存在しないワークロード、または Infrastructure as Code なしで、手動でデプロイされたワークロードを調べることができます。

コストを可視化する

推定コストの概要を含むアーキテクチャ図のコストレポートを作成します。

概念と定義

このセクションでは、主要な概念について説明し、このソリューション固有の用語を定義します。

リソース

Amazon Simple Storage Service (Amazon S3) バケットや AWS Lambda 関数などの AWS リソース。

リレーションシップ

AWS Identity and Access Management (IAM) ロールや関連する AWS Lambda 関数など、2 つのリソース間のリンク。

リソースタイプ

リソースの分類カテゴリ。常に `AWS::Lambda::Function` などの CloudFormation の命名規則に従います。

検出

AWS アカウントとリージョンのリソースとその関係をマッピングするためにソリューションが開始するプロセス。

アカウント検出モード

アカウントを見つけてソリューションに追加する方法 (AWS でのワークロード検出のウェブ UI でセルフマネージドを使用するか、AWS Organizations に委任します)。

> **Note**
>
> AWS 用語の一般的なリファレンスについては、「AWS 用語集」を参照してください。

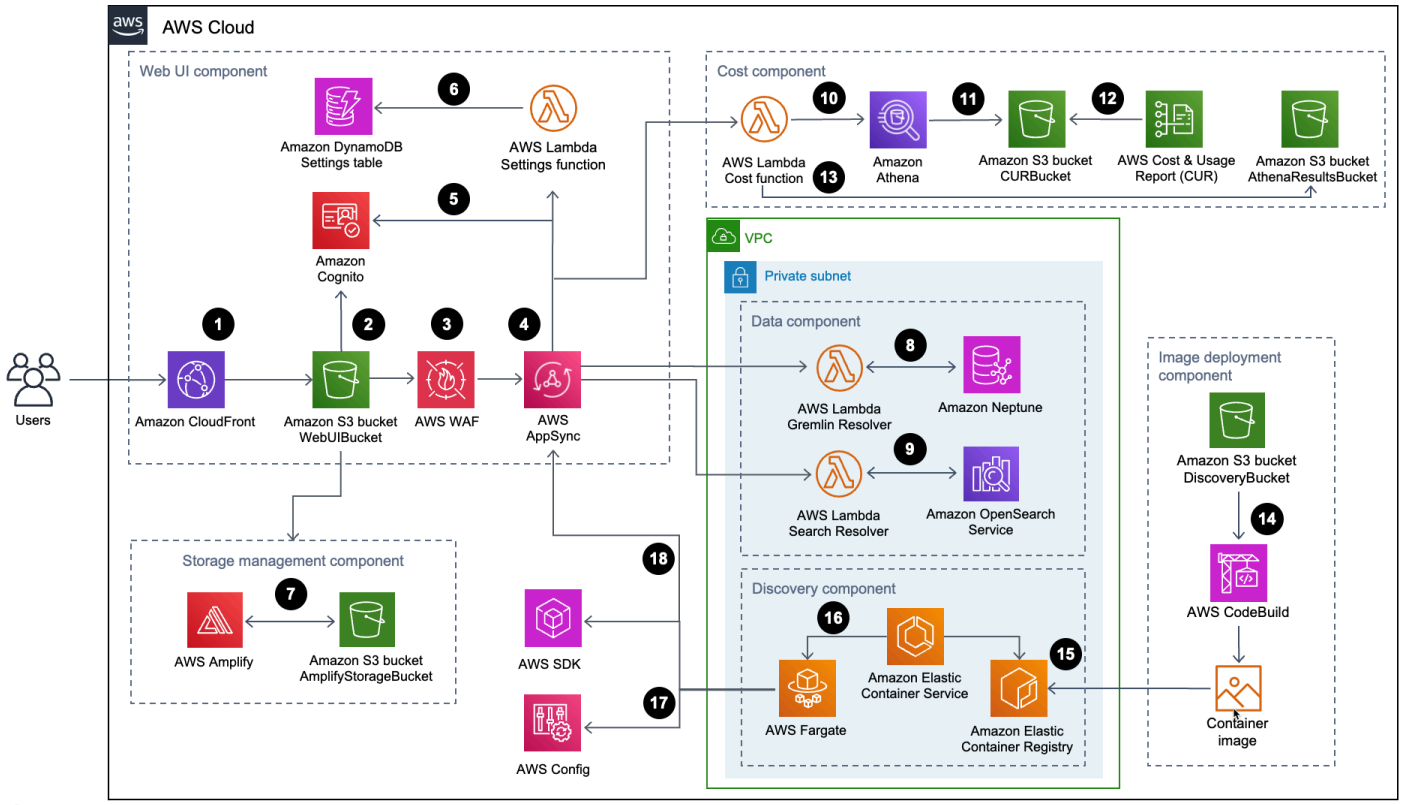
アーキテクチャの概要

このセクションでは、このソリューションで導入されるコンポーネントのリファレンス実装のアーキテクチャ図を示します。

アーキテクチャ図

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、以下に示す環境が AWS クラウドに構築されます。

AWS でのワークロード検出のアーキテクチャ



AWS CloudFormation テンプレートを使用してデプロイされたこのソリューションコンポーネントの大まかなプロセスフローは次のとおりです。

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ーを追加します。
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 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 WAF は、可用性に影響を与えたり、セキュリティを侵害したり、リソースを過剰に消費したりする可能性のある一般的なエクスプロイトやボットから AppSync API を保護します。
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関係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、AWS AppSync は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できるようにします。
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. 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。
8. 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。
9. データコンポーネントは、`Search Resolver` 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。
10. `Cost` 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。
11. Amazon Athena は AWS CUR 上でクエリを実行します。
12. AWS CUR は、レポートを `CostAndUsageReportBucket` Amazon S3 バケットに配信します。
13. `Cost` Lambda 関数は、Amazon Athena の結果を `AthenaResultsBucket` Amazon S3 バケットに保存します。
14. 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。
15. 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。
16. 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。
17. AWS Config と AWS SDK の呼び出しは、検出コンポーネントがインポートされた 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。

18AWS 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。

AWS Well-Architected の設計に関する考慮事項

このソリューションは、AWS Well-Architected フレームワークのベストプラクティスに基づいて設計されました。これにより、ユーザーは信頼性が高く、安全で、効率的で、費用対効果の高いワークロードをクラウド上で設計し運用することができます。

このセクションでは、Well-Architected Framework の設計原則とベストプラクティスがこのソリューションにどのように役立つかについて説明します。

オペレーショナルエクセレンス

このソリューションの利点となるように、オペレーショナルエクセレンスの柱の設計原則とベストプラクティスを使用してこのソリューションを設計しています。

- リソースは、CloudFormation を使用して、Infrastructure as Code として定義しました。
- このソリューションはメトリクスを Amazon CloudWatch にプッシュして、インフラストラクチャ、Lambda 関数、Amazon ECS のタスク、Amazon S3 バケット、その他のソリューションコンポーネントにオブザーバビリティを提供します。

セキュリティ

このソリューションの利点となるように、セキュリティの柱の設計原則とベストプラクティスを使用してこのソリューションを設計しています。

- Amazon Cognito がウェブ UI アプリのユーザーを認証および認可します。
- このソリューションで使用されるすべてのロールは、最小権限の原則に従います。つまり、サービスが正しく機能するために必要な最小限のアクセス許可のみが含まれます。
- 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。
- 認証情報は有効期限が短く、強力なパスワードポリシーに従っています。
- AWS AppSync のセキュリティ GraphQL ディレクティブを使用して、フロントエンドとバックエンドを呼び出すことができる操作をきめ細かく制御しています。

- 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オンにします。
- 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なります。

信頼性

このソリューションの利点となるように、信頼性の柱の設計原則とベストプラクティスを使用してこのソリューションを設計しています。

- このソリューションでは、高可用性とサービス障害からの回復を確保するために、可能な限り AWS のサーバーレスサービスを使用しています。
- すべてのコンピューティング処理には、Lambda 関数または AWS Fargate 上の Amazon ECS を使用します。
- すべてのカスタムコードは AWS SDK を使用しており、API レートクォータに達しないように、リクエストはクライアント側でスロットリングされます。

パフォーマンス効率

このソリューションの利点となるように、パフォーマンス効率の柱の設計原則とベストプラクティスを使用してこのソリューションを設計しています。

- このソリューションでは、可能な限り AWS のサーバーレスアーキテクチャを使用しています。これにより、物理サーバーを管理する運用上の負担がなくなります。
- このソリューションは、AWS Lambda、Amazon Neptune、AWS AppSync、Amazon S3、Amazon Cognito など、このソリューションで使用されている AWS サービスをサポートするすべてのリージョン起動できます。
- サポートされているリージョンで Amazon Neptune サーバーレスを使用すると、データベースのキャパシティを管理および最適化しなくても、グラフワークロードを実行して即座にスケーリングできます。
- ソリューションでは、全体を通してマネージドサービスを使用しており、リソースのプロビジョニングと管理の運用上の負担を軽減しています。

コスト最適化

このソリューションの利点となるように、コスト最適化の柱の設計原則とベストプラクティスを使用してこのソリューションを設計しています。

- AWS Fargate 上の AWS ECS ではコンピューティング専用に Lambda 関数を使用しており、使用量に応じた課金のみを行っています。
- Amazon DynamoDB は必要に応じてキャパシティをスケールするため、お支払いいただくのは使用したキャパシティに対してのみになります。

持続可能性

このソリューションの利点となるように、持続可能性の柱の設計原則とベストプラクティスを使用してこのソリューションを設計しています。

- このソリューションでは、バックエンドサービスの環境への影響を最小限に抑えるために、可能な限りマネージドサービスとサーバーレスサービスを使用しています。

アーキテクチャの詳細

このセクションでは、このソリューションを構成するコンポーネントと AWS のサービス、およびこれらのコンポーネントがどのように連携するのかについてのアーキテクチャの詳細について説明します。

[Authentication mechanism] (認証メカニズム)

AWS でのワークロード検出は、UI と AWS AppSync 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 に提供します。このトークンは、後続のすべての API リクエストで提供されます。有効な JWT が提供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。

サポート リソース

AWS でのワークロード検出が 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でのアーキテクチャ図の管理

AWS でのワークロード検出で生成されるアーキテクチャ図を、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 を使用すると、AWS でのワークロード検出は、アーキテクチャ図を Amazon S3 バケットに保存できます。次の 2 つの権限が利用可能です。

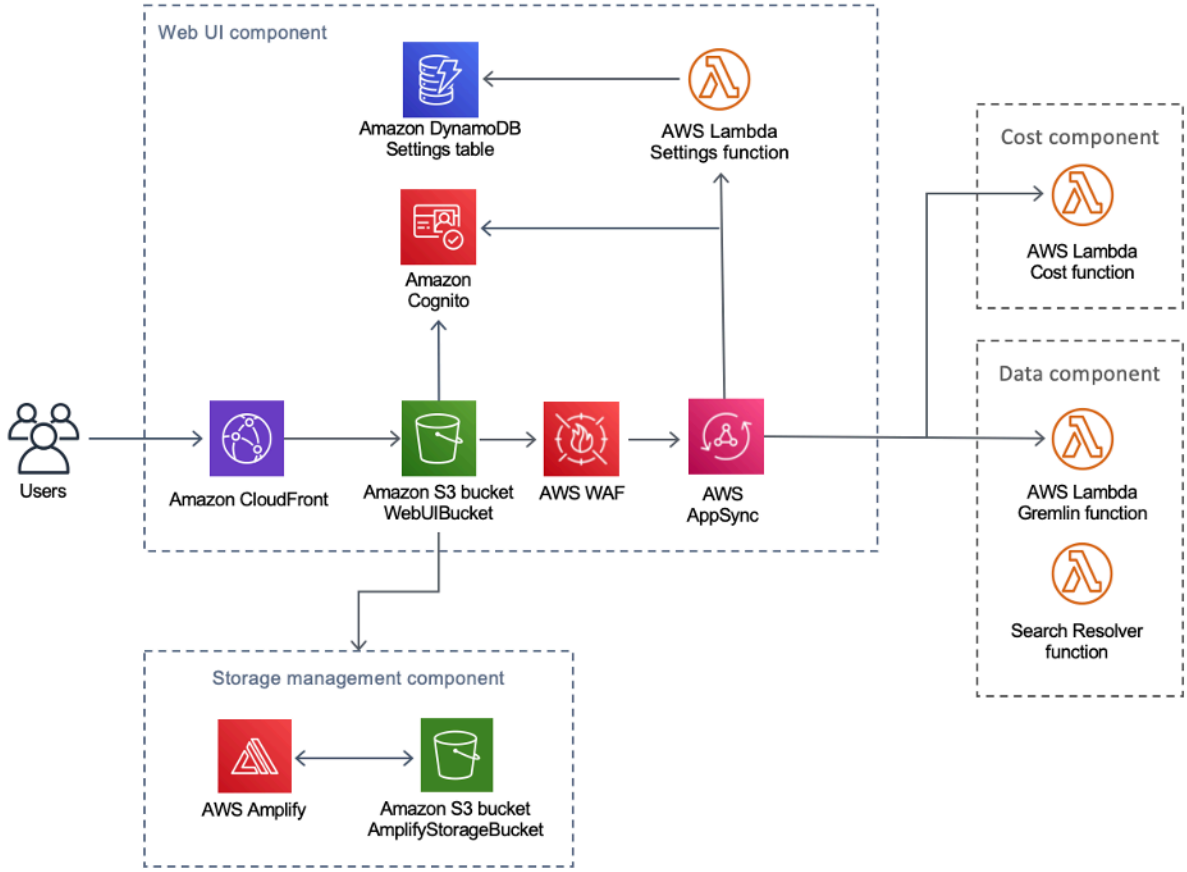
- All users - AWS でのワークロード検出のアーキテクチャ図を、デプロイ環境の AWS でのワークロード検出のユーザーに表示できるようにします。ユーザーは、それらの図をダウンロードおよび編集できます。
- You - AWS でのワークロード検出のアーキテクチャ図を、作成者のみに表示されるようにします。他のユーザーには表示されません。

ウェブ UI とストレージ管理

ウェブ UI は React を使用して開発されています。ユーザーがウェブ UI で AWS でのワークロード検出を操作できるようにフロントエンドコンソールを提供しています。

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に設定されています。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。

AWS でのワークロード検出のウェブ UI とストレージ管理コンポーネント

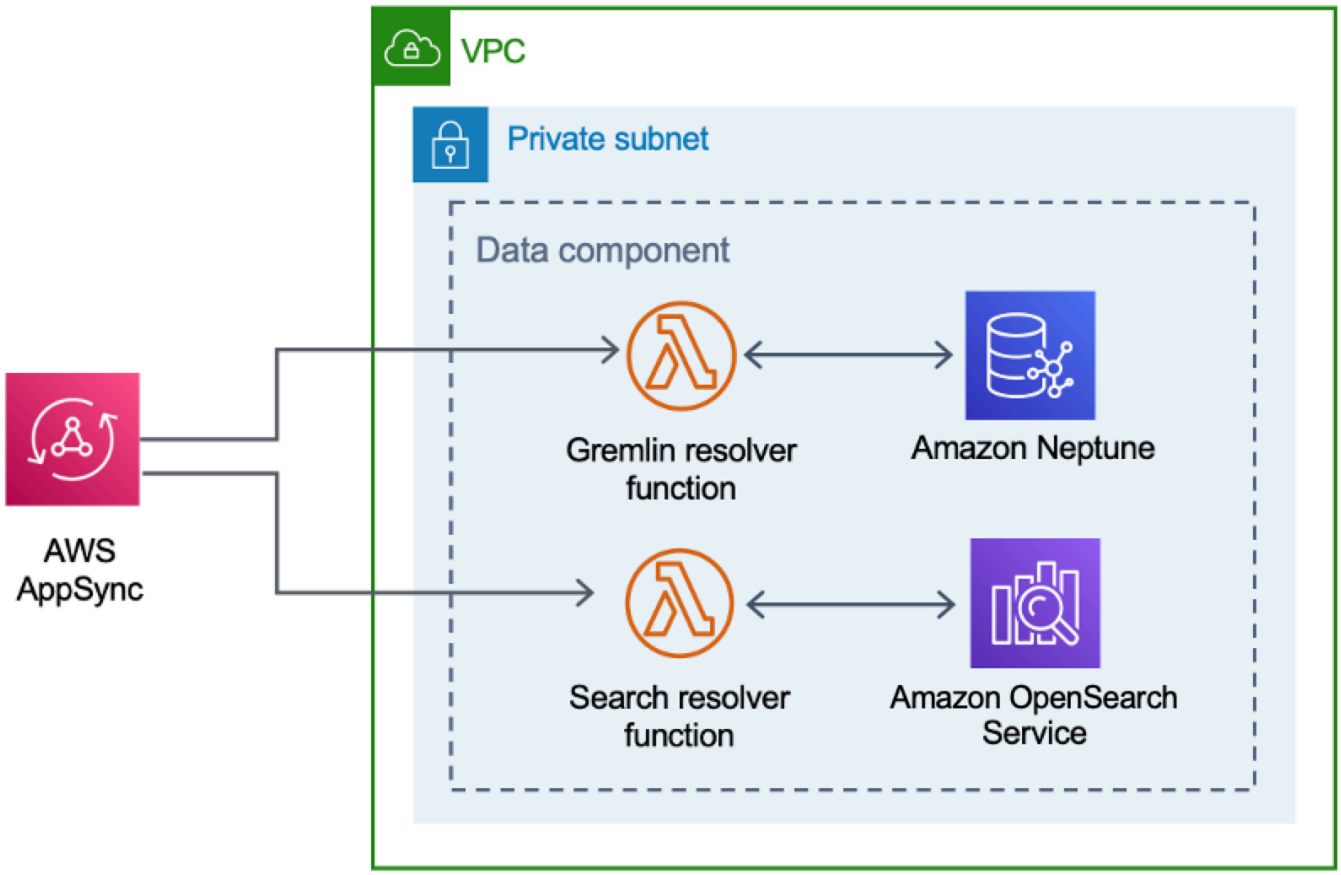


ウェブ UI リソースは `WebUIBucket` Amazon S3 バケットでホストされ、Amazon CloudFront によって配信されます。AWS Amplify は、AWS AppSync および Amazon S3 への統合をシンプルにする抽象化レイヤーを提供します。

このソリューションでは AWS AppSync を使用して、インポートされたリージョンの管理など、AWS でのワークロード検出で利用できるさまざまな設定とのやり取りを容易にします。AWS AppSync は、`Settings` AWS Lambda 関数を使用して、新しいアカウントや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

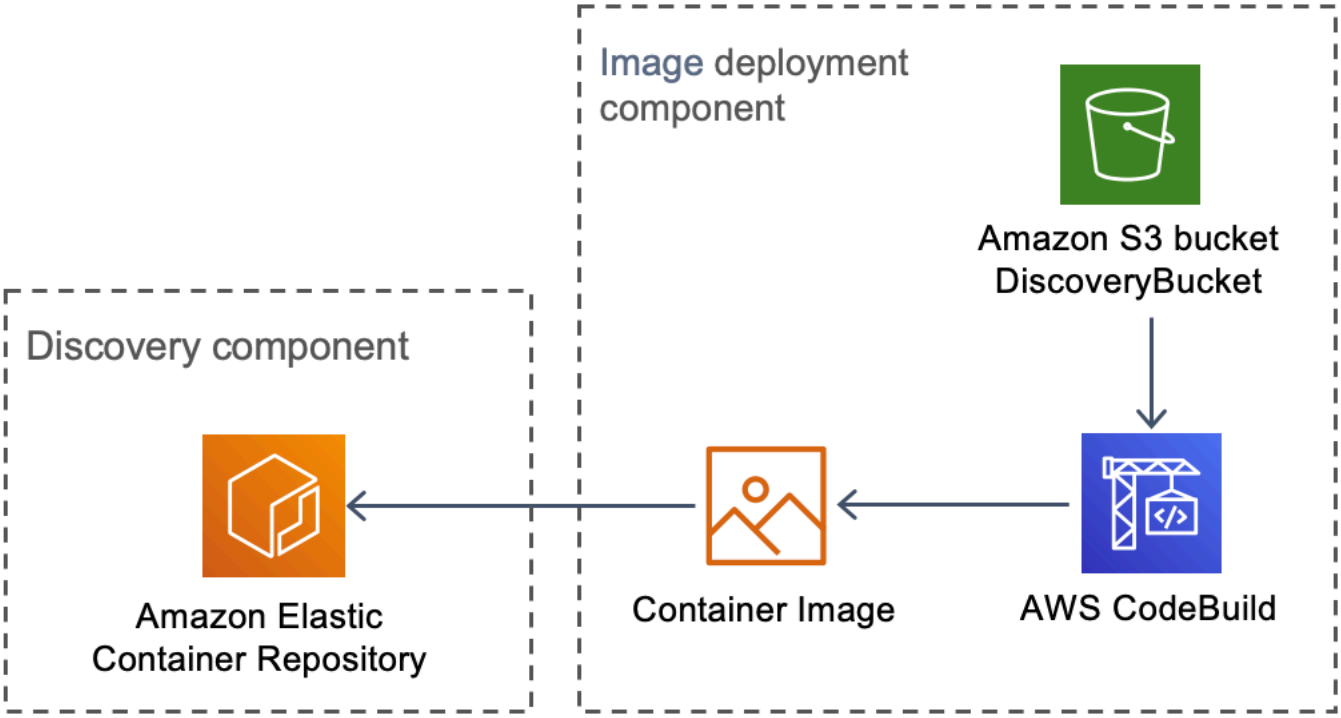
AWS でのワークロード検出のデータコンポーネント



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 Lambda 関数のいずれかが呼び出されます。これらの関数はリクエストを処理して、Amazon Neptune または 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 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ます。次に、この API はデータベースへのアクセスを提供する IAM ロールを使用して認証されます。

イメージデプロイコンポーネント



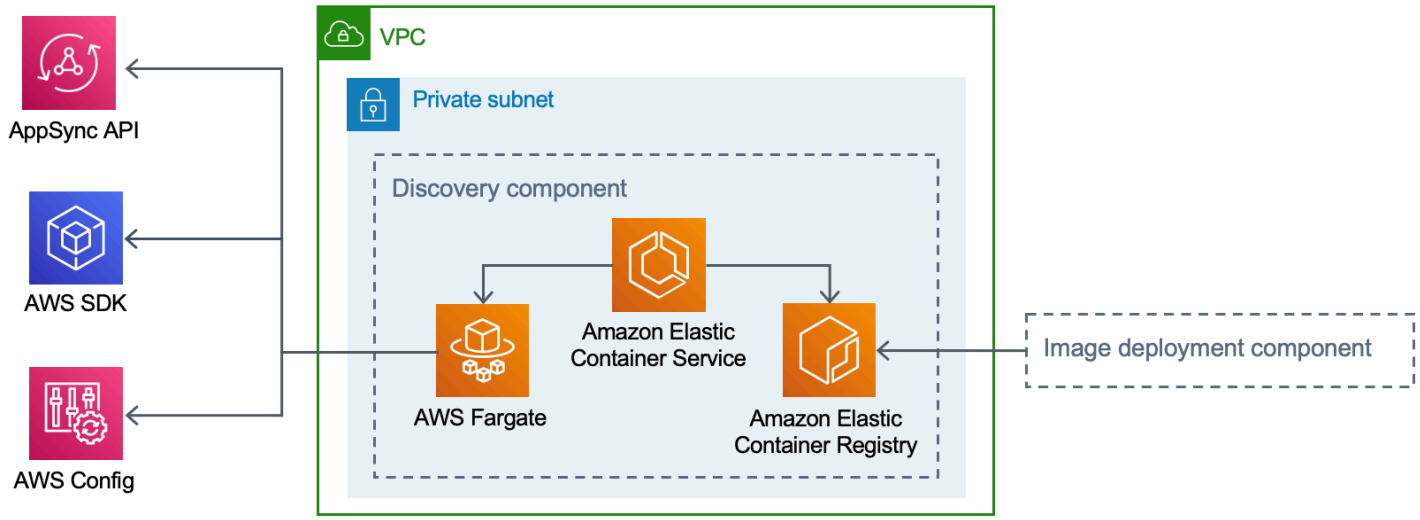
AWS でのワークロード検出のイメージデプロイコンポーネント

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するコンテナイメージを作成します。`DiscoveryBucket` と Amazon S3 バケットがコードをホストし、コンテナイメージを作成して 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 にクエリを実行し、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。

AWS でのワークロード検出の検出コンポーネント



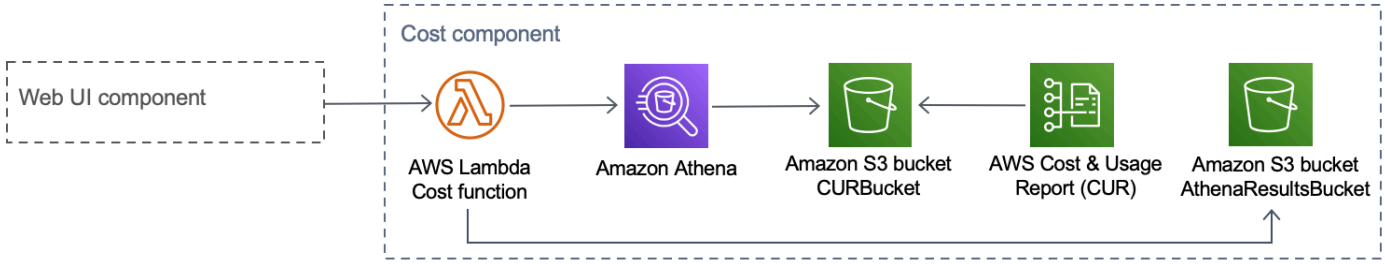
このソリューションでは、Amazon ECR からダウンロードされたコンテナイメージを使用して AWS Fargate タスクを実行するように Amazon ECS を設定します。AWS Fargate タスクは 15 分間隔で実行されるようにスケジュールされています。収集されたリソースの関係性のデータは、Amazon Neptune グラフデータベースと Amazon OpenSearch Service に保存されます。

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. Fargate タスクは、AWS Config、AWS API の describe 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. 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 コールから受信したものとの差分を計算します。
4. Fargate タスクは、リクエストを 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

AWS でのワークロード検出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 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。

このソリューションで使用している AWS のサービス

AWS のサービス	説明
AWS AppSync	コア。このソリューションは AppSync を使用して、ウェブ UI が消費するサーバーレス GraphQL API を提供します。
Amazon CloudFront	コア。このソリューションでは、Amazon S3 バケットをオリジンとして CloudFront を使用します。これにより、Amazon S3 バケットへのアクセスが制限され、パブリックにアクセスできなくなり、バケットからの直接アクセスが防止されます。
AWS Config	コア。このソリューションでは、ソリューションが検出するリソースと関係のプライマリデータソースとして AWS Config を使用します。

AWS のサービス	説明
Amazon OpenSearch Service	コア。このソリューションは、Amazon OpenSearch Service を使用して、アプリケーションのモニタリング、ログ分析、オブザーバビリティを行います。
Amazon DynamoDB	コア。このソリューションでは、DynamoDB を使用してソリューションの設定データを保存します。
Amazon Elastic Container Service (ECS)	コア。このソリューションでは、Amazon ECS を使用して、AWS アカウント内のリソースと関係を検出するタスクの実行を調整します。
AWS Fargate	コア。このソリューションでは、検出タスクのコンピューティングレイヤーとして Amazon ECS 上の AWS Fargate を使用します。
AWS Lambda	コア。このソリューションでは、Node.js と Python のランタイムを含むサーバーレスの Lambda 関数を使用して API コールを処理します。
Amazon Neptune	コア。このソリューションでは、ソリューションが検出するリソースと関係のプライマリデータストアとして Neptune を使用します。
Amazon Simple Storage Service	コア。このソリューションでは、フロントエンドとバックエンドのストレージ目的で Amazon S3 を使用します。
Amazon CloudWatch	サポート。このソリューションでは、CloudWatch を使用して、自動化されたケースでリアルタイムのログ、メトリクス、イベントデータを収集して可視化します。さらに、デプロイしたソリューションのリソース使用状況とパフォーマンスの問題を監視することもできます。

AWS のサービス	説明
AWS CodeBuild	サポート。このソリューションでは、CodeBuild を使用して、検出タスクのコードを含む Docker コンテナを構築し、フロントエンドのアセットを Amazon S3 にデプロイします。
Amazon Cognito	サポート。このソリューションは、Cognito ユーザープールを使用して、ソリューションのウェブ UI にアクセスするユーザーを認証および承認します。
AWS Systems Manager	サポート。このソリューションでは、AWS Systems Manager を使用して、アプリケーションレベルのリソース監視活動と、リソース運用とコストデータの可視化を行います。
Amazon Virtual Private Cloud	サポート。このソリューションでは、VPC を使用して Neptune データベースと OpenSearch データベースを起動します。
AWS WAF	サポート。このソリューションでは、AWS WAF を使用して、可用性に影響を与えたり、セキュリティを侵害したり、リソースを過剰に消費したりする可能性のある一般的なエクスプロイトやボットから AppSync API を保護します。
Amazon Athena	オプション。このソリューションでは、コスト機能が有効になっている場合に、Athena を使用して、コストと使用状況レポートをクエリします。

デプロイを計画する

このセクションでは、このソリューションをデプロイする前に、リージョン、コスト、セキュリティ、その他の考慮事項について説明します。

サポートしている AWS リージョン

このソリューションでは Amazon Cognito サービスを使用していますが、現在すべての AWS リージョンで利用できるわけではありません。リージョン別の AWS サービスの最新情報については、AWS リージョン別のサービスのリストを参照してください。

AWS でのワークロード検出は、次の AWS リージョンで利用できます。

リージョン名	
米国東部 (バージニア北部)	カナダ (中部)
米国東部 (オハイオ)	欧州 (ロンドン)
米国西部 (オレゴン)	欧州 (フランクフルト)
アジアパシフィック (ムンバイ)	欧州 (アイルランド)
アジアパシフィック (ソウル)	欧州 (パリ)
アジアパシフィック (シンガポール)	欧州 (ストックホルム)
アジアパシフィック (シドニー)	南米 (サンパウロ)
アジアパシフィック (東京)	

AWS でのワークロード検出は、次の AWS リージョンでは利用できません。

リージョン名	利用できないサービス
AWS GovCloud (米国東部)	AWS AppSync
AWS GovCloud (米国西部)	AWS AppSync

リージョン名	利用できないサービス
中国 (北京)	Amazon Cognito
中国 (寧夏)	Amazon Cognito

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 サービスのコストは、お客様の負担となります。この改訂の時点で、米国東部 (バージニア北部) リージョンで、単一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。

> **Note**
>
> AWS でのワークロード検出を 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。次表の例に示す AWS のサービスは、月単位で請求されます。

AWS Cost Explorer を使用して予算を作成することをお勧めします。これはコスト管理に役立ちます。料金は変更されることがあります。詳細については、このソリューションで使用する AWS のサービスごとに料金ウェブページを参照してください。

サンプルコスト表

オプション 1: 単一インスタンスのデプロイ (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、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、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	1 時間あたりのコスト [USD]	月額コスト [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 ゲートウェイ)	該当なし	0.090 USD	65.7 USD
AWS Config	該当なし	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)	該当なし	0.02 USD	12.01 USD
Total		0.586 USD	425.19 USD

オプション 2: 複数のインスタンスのデプロイ

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、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、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	1 時間あたりのコスト	月額コスト [USD]
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 ゲートウェイ)	該当なし	0.090 USD	65.7 USD

AWS のサービス	インスタンスタイプ	1 時間あたりのコスト	月額コスト [USD]
AWS Config	該当なし	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)	該当なし	0.02 USD	12.01 USD
Total		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。

> **Important**
>
> 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。

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と、お客様と AWS の間でセキュリティ上の責任が分担されます。この責任共有モデルにより、ホストオペレーティングシステムと仮想化レイヤーからサービスが運用されている施設の物理的なセキュリティに至るまで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するのに役立ちます。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。

リソースアクセス

IAM ロール

IAM ロールを使用すると、AWS クラウドのサービスとユーザーに、きめ細かなアクセスポリシーとアクセス許可を割り当てることができます。AWS でのワークロード検出を実行して、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に必要なコンポーネントへのアクセスを認証し、一時的で強固な認証情報をしてアクセスを認証するために使用されます。

ネットワークアクセス

Amazon VPC

AWS でのワークロード検出は 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。

デプロイが完了し起動したら、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 によって配布される Amazon S3 バケットにホストされるウェブコンソール UI をデプロイします。オリジンアクセスアイデンティティ機能を使用するとで、この Amazon S3 バケットのコンテンツには CloudFront からのみアクセスできます。詳細については、「Amazon CloudFront デベロッパーガイド」の「Amazon S3 オリジンへのアクセスの制限」を参照してください。

CloudFront は追加のセキュリティ対策を有効にして、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します。詳細については、「CloudFront レスポンスでの HTTP ヘッダーの追加または削除」を参照してください。

このソリューションでは、デフォルトの CloudFront 証明書を使用しており、サポートされる最小のセキュリティプロトコルは TLS v1.0 です。TLS v1.2 または TLS v1.3 の使用を必ず適用するには、デフォルトの CloudFront 証明書の代わりにカスタム 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、認証と許可は IAM と Amazon Cognito を使用して実装されます。これにより、ユーザーがウェブ UI で正常に認証されたときに Amazon Cognito が提供する JWT を使用します。

AWS Lambda

デフォルトでは、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。サービスのやり取りは、必要最小限の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、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 Lambda 関数に制限されています。

OpenSearch Service クラスターは、ノード間の暗号化を有効にして構築されており、既存の OpenSearch Service のセキュリティ機能の上にデータ保護の層を追加します。

クォータ

サービスクォータ (制限とも呼ばれます) は、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です。

このソリューション内の AWS サービスのクォータ

このソリューションに実装されている各サービスに十分なクォータがあることを確認してください。詳細については、「AWS のサービスクォータ」を参照してください。

次のリンクを使用して、そのサービスのページに移動します。ページを切り替えずに、ドキュメント内のすべての AWS サービスのサービスクォータを表示するには、この PDF の「Service endpoints and quotas」ページの情報を参照してください。

Amplify	Amazon ECR

Athena	Lambda
CloudFront	OpenSearch Service
Cognito	Neptune
設定	Amazon S3
Amazon ECS	

AWS CloudFormation のクォータ

ご使用の AWS アカウントには AWS CloudFormation のクォータがあり、このソリューションでスタックを起動する際に注意する必要があります。これらのクォータを理解することで、このソリューションを正常にデプロイできなくなるような制限エラーを回避できます。詳細については、「AWS CloudFormation ユーザーガイド」の「CloudFormation クォータを理解する」を参照してください。

AWS Lambda のクォータ

ご使用のアカウントでは AWS Lambda 同時実行クォータは 1,000 になります。Lambda を実行して使用している他のワークロードがあるアカウントでソリューションを使用する場合は、このクォータを適切な値に設定します。この値は調整可能です。詳細については、「AWS Lambda ユーザーガイド」の「AWS Lambda クォータ」を参照してください。

> **Note**
>
> このソリューションでは、ソリューションがデプロイされるアカウントに同時実行クォータから 150 の実行が利用可能である必要があります。そのアカウントで利用可能な実行数が 150 未満の場合、CloudFormation デプロイは失敗します。

Amazon VPC クォータ

ご使用の AWS アカウントには、5 つの VPC と 2 つの Elastic IP (EIP) を含めることができます。他の VPC または EIP のアカウントでソリューションを使用する場合は、このソリューションを正常にデプロイできない可能性があります。このクォータに達するリスクがある場合は、「スタックを起動する」セクションの手順に従って VPC を設定することで、デプロイにご自分の VPC を使用できま

す。詳細については、「Amazon VPC ユーザーガイド」の「Amazon VPC クォータ」を参照してください。

デプロイするアカウントを選択する

AWS Organizations に AWS でのワークロード検出をデプロイする場合は、StackSets とマルチリージョン AWS Config 機能が有効になっている委任管理者アカウントにこのソリューションをインストールする必要があります。

AWS Organizations を使用していない場合は、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場所を提供します。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。

ソリューションをデプロイする

このソリューションは、AWS CloudFormation テンプレートとスタックを使用してデプロイを自動化します。CloudFormation テンプレートは、このソリューションに含まれる AWS リソースとそのプロパティを指定します。CloudFormation スタックは、テンプレートに記述されているリソースをプロビジョニングします。

デプロイプロセスの概要

> **Note**
>
> 既に AWS でのワークロード検出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、「ソリューションを更新する」を参照してください。

このセクションのステップバイステップの手順に従って、ソリューションを設定してアカウントにデプロイします。

デプロイ時間: 約 30 分

このソリューションを起動する前に、コスト、アーキテクチャ、ネットワークセキュリティなど、このガイドで説明されている考慮事項を確認してください。

> **Important**
>
> このソリューションには、匿名化された運用メトリクスを AWS に送信するオプションが含まれています。AWS で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 プライバシー通知が適用されます。

前提条件

デプロイパラメータの詳細を収集する

AWS でのワークロード検出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 OpenSearch Service クラスターを作成します。ただし、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:

1. このソリューションをデプロイする予定のアカウントの Identity and Access Management (IAM) コンソールにサインインします。
2. [Search (検索)] ボックスに、`AWSServiceRoleForAmazonOpenSearchService` を入力します。
3. 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では、AWS Config を使用してリソース設定の大部分を収集します。このソリューションをデプロイしたり、新しい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 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をデプロイするか、AWS でのワークロード検出にインポートするリージョンでこのコマンドを実行します。

次のコマンドを入力します。

```
aws configservice get-status
```

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

出力:

```
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
```

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リージョンのバッチにこの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で検出可能にする場合は、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:

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。

VPC 設定を検証する

既存の VPC にデプロイする場合は、プライベートサブネットが AWS のサービスにリクエストをルーティングできることを確認します。

既存の VPC にソリューションをデプロイするオプションを選択する場合は、AWS でのワークロード検出の Lambda 関数とご自分の VPC のプライベートサブネットで実行されている Amazon ECS タスクが他の AWS のサービスに接続できることを確認する必要があります。これを有効にする標準的な方法は、NAT ゲートウェイを使用することです。次のコードサンプルに示すように、ご自分のアカウントで NAT ゲートウェイを一覧表示できます。

```
aws ec2 describe-route-tables --filters Name=association.subnet-id,Values=<private-
subnet-id1>,<private-subnet-id2> --query 'RouteTables[].Routes[].NatGatewayId'
```

出力:

```
[
    "nat-1111111111111111",
    "nat-2222222222222222"
]
```

> **Note**
>
> 返される結果が 2 つ未満の場合、サブネットに正しい数の NAT ゲートウェイがありません。

ご自分の VPC に NAT ゲートウェイがない場合は、NAT ゲートウェイをプロビジョニングするか、「AWS API」セクションに記載されているすべての AWS サービスに VPC エンドポイントがあることを確認する必要があります。

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 クラウドでの AWS でのワークロード検出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 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。

View template

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、「このソリューションで使用している AWS のサービス」セクションに記載しているコアとサポートのサービスがデプロイされますが、特定のニーズに合わせてテンプレートをカスタマイズできます。

> **Note**
>
> 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。

スタックを起動する

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。

デプロイ時間: 約 30 分

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。

   Launch solution

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) リージョンで起動します。別の AWS リージョンで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でリージョンセレクターを使用します。

   > **Note**
   >
   > 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、「サポートしている AWS 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表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、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	*<####>*	最初のユーザーを作成する E 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。
ApiAllowListedRanges	`0.0.0.0/1,128.0.0.0/1`	AppSync GraphQL API へのアクセスを管理するための CIDR をカンマで区切ったリスト。インターネット全体を許可するには、0.0.0.0/1、128.0.0.0/1 を使用します。特定の CIDR へのアクセスを制限する場合は、プライベートサブネットで実行されている検出プロセス ECS タスクがインターネットにアクセスできるようにする NAT ゲートウェイの IP アドレス (および /32 のサブネットマスク) も含める必要があります。注: この許可リストは WebUI へのアクセスを管理するものではなく、GraphQL API のみを管理します。

パラメータ	デフォルト	説明
CreateNeptuneReplica	No	別のアベイラビリティーゾーンで 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`	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。

パラメータ	デフォルト	説明
OpensearchMultiAz	No	複数のアベイラビリティーゾーンにまたがる 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。
CrossAccountDiscovery	SELF_MANAGED	アカウントのインポートを AWS でのワークロード検出と AWS Organizations のどちらで管理するかを選択します。ここには、SELF_MANAGED または AWS_ORGANIZATIONS が表示されます。
OrganizationUnitId	<オプション入力>	ルート組織のユニット ID。このパラメータは、CrossAccountDiscovery が AWS_ORGANIZATIONS に設定されている場合にのみ使用します。
AccountType	DELEGATED_ADMIN	AWS でのワークロード検出をインストールする AWS Organizations アカウントのタイプ。このパラメータは、CrossAccountDiscovery が AWS_ORGANIZATIONS に設定されている場合にのみ使用します。詳細については、「デプロイするアカウントの選択」を参照してください。

パラメータ	デフォルト	説明
ConfigAggregatorName	<オプション入力>	使用する AWS Organizations 全体の Config アグリゲータ。このアグリゲータと同じアカウントとリージョンにソリューションをインストールする必要があります。このパラメータを空白のままにすると、新しいアグリゲータが作成されます。このパラメータは、CrossAccountDiscovery が `AWS;_ORGANIZATIONS` に設定されている場合にのみ使用します。
CpuUnits	`1 vCPU`	検出プロセスが実行される Fargate タスクに割り当てる CPU の数。
「メモリ」	`2048`	検出プロセスが実行される Fargate タスクに割り当てるメモリの量。
DiscoveryTaskFrequency	`15mins`	検出プロセスの ECS タスクを実行するたびに実行される時間間隔。
MinNCUs	`1`	Neptune クラスターに設定する Neptune キャパシティユニット (NCU) の最小数 (MaxNCUs 以下でなければなりません)。DB インスタンスタイプが `db.serverless` の場合は、必須です。

パラメータ	デフォルト	説明
MaxNCUs	128	Neptune クラスターに設定する NCU の最大数。(MinNCUs 以上でなければなりません)。DB インスタンスタイプが `db.serverless` の場合は、必須です。
VpcId	<オプション入力>	ソリューションが使用する既存 VPC の ID。このパラメータを空白のままにすると、新しい VPC がプロビジョニングされます。
VpcCidrBlock	<オプション入力>	VpcId パラメータによって参照される VPC の VPC CIDR ブロック。このパラメータは、VpcId パラメータが設定されている場合にのみ使用できます。
PrivateSubnet0	<オプション入力>	使用したいプライベートサブネット。このパラメータは、VpcId パラメータが設定されている場合にのみ使用できます。
PrivateSubnet1	<オプション入力>	使用したいプライベートサブネット。このパラメータは、VpcId パラメータが設定されている場合にのみ使用できます。
UsesCustomIdentity	No	SAML や OIDC などのカスタム ID プロバイダーを使用するかどうかの確認。

パラメータ	デフォルト	説明
CognitoCustomDomain	<オプション入力>	アプリケーションのサインアップページおよびサインインページをホストする Amazon Cognito カスタムドメインのドメインプレフィックス。カスタム IdP を使用していない場合は空のままにします。そうでない場合は、小文字、数字、ハイフンのみを含める必要があります。
CognitoAttributeMapping	<オプション入力>	標準およびカスタム Cognito ユーザープール属性への IdP 属性のマッピング。カスタム IdP を使用していない場合は空のままにします。そうでない場合は、有効な JSON 文字列である必要があります。
IdentityType	<オプション入力>	使用する ID プロバイダーのタイプ (`Google`、`SAML`、または `OIDC`)。カスタム IdP を使用していない場合は空のままにします。
ProviderName	<オプション入力>	ID プロバイダーの名前。カスタム IdP を使用していない場合は空のままにします。
GoogleClientId	<オプション入力>	使用する Google クライアント ID。IdentityType が `Google` に設定されている場合にのみ使用するパラメータ。

パラメータ	デフォルト	説明
GoogleClientSecret	<オプション入力>	使用する Google クライアントシークレット。IdentityType が `Google` に設定されている場合にのみ使用するパラメータ。
SAMLMetadataURL	<オプション入力>	SAML ID プロバイダーのメタデータ URL。IdentityType が SAML に設定されている場合にのみ使用するパラメータ。
OIDCClientId	<オプション入力>	使用する OIDC クライアント ID。IdentityType が `OIDC` に設定されている場合にのみ使用するパラメータ。
OIDCClientSecret	<オプション入力>	使用する OIDC クライアントシークレット。IdentityType が `OIDC` に設定されている場合にのみ使用するパラメータ。
OIDCIssuerURL	<オプション入力>	使用する OIDC 発行者 URL。IdentityType が `OIDC` に設定されている場合にのみ使用するパラメータ。
OIDCAttributeRequestMethod	`GET`	使用する OIDC 属性リクエストメソッド。`GET` または `POST` のいずれかである必要があります (OIDC プロバイダーを参照するか、デフォルト値を使用してください)。IdentityType が `OIDC` に設定されている場合にのみ使用するパラメータ。

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. [確認および作成] ページで、設定を確認して確定します。テンプレートが IAM リソースを作成し、特定の機能が必要であることを確認するチェックボックスを選択します。
9. [送信] を選択してスタックをデプロイします。

   AWS CloudFormation コンソールの [ステータス] 列でスタックのステータスを確認できます。約 30 分後に CREATE_COMPLETE ステータスが表示されます。

> **Note**
>
> 削除すると、このスタックはすべてのリソースを削除します。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。

デプロイ後の設定作業

AWS でのワークロード検出が正常にデプロイされたら、次のデプロイ後のタスク設定を実行します。

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 デベロッパーガイド」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。

> **Note**
>
> 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。デプロイ時にユーザーが作成され、AdminUserEmailAddress パラメータで指定されたアドレスに一時的な認証情報を記載した E メールが送信されます。

追加のユーザーを作成するには:

1. AWS Cognito コンソールにサインインします。
2. [Manage User Pools] (ユーザープールの管理) をクリックします。
3. [WDCognitoUserPool- *<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
ユーザーネーム	はい	AWS でのワークロード検出へのログインに使用するユーザー名。

フォームのフィールド	必須?	説明
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] のみを選択します。[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、ユーザーは引き続き作成されます。
仮パスワード	はい	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に初めてサインインするときに、パスワードの変更を強制されます。
電話番号	いいえ	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としてマークする] ボックスが選択されていることを確認してください。
E メール	はい	有効な E メールアドレスを入力します。[E メールアドレスを検証済みとしてマークする] 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。

> **Note**
>
> 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アカウントには、AWS でのワークロード検出を個

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にログインする

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 AWS で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サインインの認証情報を入力します。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、アカウントの復旧を完了します。

リージョンをインポートする

> **Note**
> 次のセクションは、ソリューションのアカウント検出モードがセルフマネージドの場合にのみ適用されます。AWS Organizations モードでアカウントを検出する仕組みについては、「AWS Organizations のアカウント検出モード」セクションを参照してください。

リージョンをインポートするには、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 - アカウントに 1 度デプロイされ、インポートされた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

リージョンをインポートする

これらの手順では、リージョンをインポートして 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 でのワークロード検出にサインインします。URL については、「AWS でのワークロード検出にログインする」を参照してください。
2. ナビゲーションメニューで [Accounts] を選択します。
3. [インポート] を選択します。
4. インポート方法を選択します。
   a. CSV ファイルを使用してアカウントとリージョンを追加。
   b. フォームを使用してアカウントとリージョンを追加。

CSV ファイル

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

```
"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. 関連するアカウントに CloudFormation テンプレートをデプロイします (「AWS CloudFormation テンプレートをデプロイする」セクションを参照)。
6. Global と Regional のリソーステンプレートがデプロイされたら、両方の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] を選択します。

フォーム

次のフォームを使用して、インポートするリージョンを指定します。

1. Account ID には 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 には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. インポートする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. 関連するアカウントに CloudFormation テンプレートをデプロイします (「AWS CloudFormation テンプレートをデプロイする」セクションを参照)。
8. Global と Regional のリソーステンプレートがデプロイされたら、両方の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] 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、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にインポート済みのリージョンを含むアカウントから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。リージョンが既にインポートされている場合は、「スタックをデプロイして Regional リソースをプロビジョニングする」の手順に従ってください。

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

> **Important**
>
> 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. ナビゲーションメニューから [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 [アクセス許可] で、以下を実行します。
   a. AWS Organizations を使用している場合は、[サービスマネージド型のアクセス許可] または [セルフサービス型のアクセス許可] のいずれかを選択します。詳細については、「AWS Organizations での StackSet の使用」を参照してください。
   b. AWS Organizations を使用していない場合は、StackSets の前提条件の手順に従うときに使用する IAM の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。
5. [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] を選択します。`global-resources.template` ファイル (CSV ファイルまたはフォームでリージョンをインポートした際にダウンロード済み) を選択し、[次へ] を選択します。
6. [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 AWS STS クォータ」を参照してください。
7. [パラメータ] で、このソリューションのテンプレート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。この値はデフォルトのま

フィールド名	デフォルト	説明
		まにしておく必要があります。

1. [次へ] を選択します。
2. [StackSet オプションの設定] ページで、[次へ] を選択します。
3. [デプロイオプションの設定] ページの [アカウント] で、[アカウント番号] にアカウントロールをデプロイするアカウント ID を入力します。
4. [リージョンの指定] で、スタックをインストールするリージョンを 1 つ選択します。
5. [デプロイオプション] で [並行] を選択し、[次へ] を選択します。
6. [確認] ページで、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確認するボックスにチェックを入れます。
7. [Submit] を選択してください。

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

> **Important**
> 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リージョンとインストールされていない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. ナビゲーションメニューから [StackSets] を選択します。
3. [Create StackSet] (StackSet の作成) を選択します。
4. テンプレートの選択ページの [アクセス許可] で、以下を実行します。
   a. AWS Organizations を使用している場合は、[サービスマネージド型のアクセス許可] または [セルフサービス型のアクセス許可] のいずれかを選択します。詳細については、「AWS Organizations での StackSet の使用」を参照してください。

b. AWS Organizations を使用していない場合は、StackSets の前提条件の手順に従うときに使用する IAM の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。

5. [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] を選択します。`regional-resources.template` ファイル (CSV ファイルまたはフォームでリージョンをインポートした際にダウンロード済み) を選択し、[次へ] を選択します。
6. [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 AWS STS クォータ」を参照してください。
7. [パラメータ] で、このソリューションのテンプレート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。この値はデフォルトのままにしておく必要があります。
AggregationRegion	デプロイリージョン	最初にデプロイされたリージョン。この値はデフォルトのままにしておく必要があります。
AlreadyHaveConfigSetup	No	リージョンで既に AWS Config が有効になっているか確認します。AWS Config がこの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。

1. [次へ] を選択します。
2. [StackSet オプションの設定] ページで、[次へ] を選択します。

3. [デプロイオプションの設定] ページの [アカウント] で、[アカウント番号] にアカウントロールをデプロイするアカウント ID を入力します。
4. [リージョンの指定] で、スタックをインストールするリージョンを 1 つ選択します。これにより、手順 6 で入力したすべてのアカウントのこれらのリージョンにスタックがインストールされます。
5. [デプロイオプション] で [並行] を選択し、[次へ] を選択します。
6. [確認] ページで、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確認するボックスにチェックを入れます。
7. [Submit] を選択してください。

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、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にインポート済みのリージョンを含むアカウントから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、`global-resources.template` ファイル (CSV ファイルまたはフォームでリージョンをインポートした際にダウンロード済み) を選択し、[次へ] を選択します。
5. [スタックの詳細を指定] ページで、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 AWS 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
スタック名	`workload-discovery`	この AWS CloudFormation スタックの名前。

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。この値はデフォルトのままにしておく必要があります。

1. [次へ] を選択します。
2. 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確認するボックスを選択します。
3. [スタックの作成] を選択してください。

新しいリージョンは、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)

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、`regional-resources.template` ファイル (CSV ファイルまたはフォームでリージョンをインポートした際にダウンロード済み) を選択し、[次へ] を選択します。
5. [スタックの詳細を指定] ページで、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 AWS 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	ソリューションの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	ソリューションのデプロイリージョン	最初にデプロイされたリージョン。デフォルトのままにしてください。
AlreadyHaveConfigSetup	No	リージョンで既に AWS Config が有効になっているか確認します。AWS Config がこの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。

1. [次へ] を選択します。
2. 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確認するボックスを選択します。
3. [スタックの作成] を選択してください。

新しいリージョンは、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)

リージョンが正しくインポートされたことを確認する

1. ソリューションのウェブ UI にサインインします (または、既に読み込まれている場合はページを更新します)。URL については、「AWS でのワークロード検出にログインする」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。

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。

> **Note**
>
> Last Scanned 列が 30 分以上空白のままになっている場合は、「検出コンポーネントのデバッグ」を参照してください。

コスト機能の設定

コスト機能を使用するには、AWS の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。次の手順を行ってください。

1. スケジュールされた CUR を設定します。
2. Amazon S3 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の外部にある場合)。

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アカウントの Billing コンソールにサインインします。
2. ナビゲーションメニューの [Billing] で、[Cost & usage reports] を選択します。
3. [レポートを作成] 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。

   > **Note**
   >
   > 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] ボックスを選択します。

   > **Note**
   >
   > 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 0] 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. レポートのプレフィックスパスを `aws-perspective` 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。

> **Note**
>
> 注記レポートが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アカウントの Billing コンソールにサインインします。
2. ナビゲーションメニューの [コスト管理] で、[コストと使用状況レポート] を選択します。
3. [レポートを作成] 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。

   > **Note**
   >
   > 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] ボックスにチェックを入れます。

> **Note**
> 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 0] 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. レポートのプレフィックスパスを `aws-perspective` 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。

> **Note**
> 注記レポートが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。

レプリケーションの設定

デプロイ中に作成された 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、ソリューションが Amazon Athena でバケットをクエリ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 を含む Amazon S3 コンソールの AWS アカウントに サインインします。
2. CUR の設定時に作成された Amazon S3 バケットを選択します。詳細については、「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 を参照してください。

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、説明的なルール 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のデプロイ時に作成された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] ボックスを選択します。
8. IAM ロールで、[新しいロールの作成] を選択します。

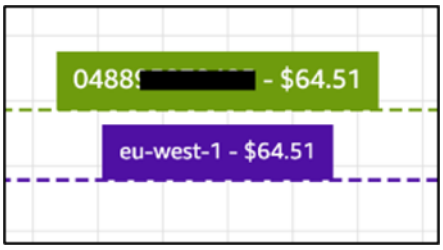
> **Note**
>
> レプリケーションロールが既に存在している場合があります。これを選択して、必要な S3 レプリケーションロールアクションがあることを確認できます。

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サインインし、S3 のサービスページに移動して、`CostAndUsageReportBucket` 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。S3 バケットのポリシー例については、「Cost Bucket のレプリケーションポリシー」を参照してください。

> **Note**
>
> 複数の AWS アカウントから CUR をレプリケートする場合。送信先バケット (AWS でのワークロード検出のアカウント内) のバケットポリシーに、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、「Cost Bucket のレプリケーションポリシー」を参照してください。

レポートが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。



S3 バケットのライフサイクルポリシーの編集

このソリューションは、デプロイ時に、次の 2 つのバケットにライフサイクルポリシーを設定します。

- `CostAndUsageReportBucket`
- `AccessLogsBucket`

> **Important**
>
> これらのライフサイクルポリシーは、90 日後にこれらの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、所有する内部ポリシーに合わせることができます。

ソリューションのモニタリング

このソリューションでは、myApplications と CloudWatch AppInsights を使用して、AWS でのワークロード検出のデプロイをモニタリングできます。

myApplications

myApplications は、コンソールホームの拡張機能であり、AWS でのアプリケーションのコスト、ヘルス、セキュリティ体制、パフォーマンスの管理とモニタリングに役立ちます。AWS マネジメントコンソールの 1 つのビューから、アカウント内のすべてのアプリケーション、すべてのアプリケーションの主要なメトリクス、および複数のサービスコンソールのコスト、セキュリティ、運用のメトリクスとインサイトの概要にアクセスできます。

AWS でのワークロード検出用の myApplications ダッシュボードを表示するには:

1. AWS マネジメントコンソールにサインインします。
2. 左側のサイドバーで [myApplications] を選択します。
3. 検索バーに `workload-discovery` を入力してアプリケーションを見つけます。
4. アプリケーションを選択します。

CloudWatch AppInsights

CloudWatch Application Insights は、アプリケーションリソースと技術スタック全体で重要なメトリクス、ログ、アラームを特定し設定することで、アプリケーションのモニタリングに役立ちます。メトリクスとログを継続的にモニタリングし、異常やエラーを検出して相互に関連付けます。トラブルシューティングに役立てるために、検出した問題の自動ダッシュボードを作成します。このダッシュボードには、相互に関連付けられた異常とログエラー、さらに根本原因を示唆する追加のインサイトが示されます。

AWS でのワークロード検出用の CloudWatch AppInsights ダッシュボードを表示するには:

1. CloudWatchコンソールにサインインします。
2. 左側のサイドバーで、インサイト、Application Insights を選択します。
3. [アプリケーション] タブを選択します。
4. 検索バーに `workload-discovery` と入力してダッシュボードを検索します。

5. ダッシュボードを選択します。
6. アプリケーションを選択します。

ソリューションを更新する

> **Important**
>
> AWS でのワークロード検出の v1.x.x から v2.x.x へのアップデートはサポートされていません。v2.x.x をインストールする前に、このソリューションの 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. 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、[現在の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] ページの [スタック障害オプション] で、[プロビジョニングの失敗時の動作] ラジオボタンが [すべてのスタックリソースをロールバック] に設定されていることを確認します。
9. [次へ] を選択します。
10. [レビュー] ページで、設定を確認します。テンプレートが IAM リソースを作成し、特定の機能が必要であることを確認するボックスを選択します。
11. [スタックの更新] を選択してスタックをデプロイします。

> **Note**
>
> このソリューションをセルフマネージドのアカウント検出モードでデプロイした場合は、「リージョンをインポートする」セクションの手順に従って、デプロイしたグローバルリソースを更新する必要があります。

トラブルシューティング

既知の問題解決には、既知のエラーを軽減するための手順が記載されています。これらの手順で問題が解決しない場合は、AWS サポートに問い合わせるセクションで、このソリューションに関する AWS サポートのケースを開く手順を参照してください。

既知の問題解決

AWS でのワークロード検出のデプロイ中およびデプロイ後のフェーズでは、いくつかの一般的な設定エラーが発生する可能性があります。

> **Note**
>
> 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。AWS でのワークロード検出のデプロイ後の設定に関するドキュメントにも、追加のトラブルシューティングのヘルプが記載されています。

配信チャネル設定のエラー

問題: メインの AWS CloudFormation テンプレートをデプロイするときに次のエラーが発生します。

```
Failed to put delivery channel '<stack-name>-DiscoveryImport-<ID-string>-
DeliveryChannel-<ID-string>' because the maximum number of delivery channels:
 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code:
 MaxNumberOfDeliveryChannelsExceededException; Request ID: 4edc54bc-8c85-4925-
b99d-7ef9c73215b3; Proxy: null)
```

理由: ソリューションは、AWS Config が既に有効になっているリージョンにデプロイされています。

解決策: 前提条件セクションの手順に従って、CloudFormation パラメータ AlreadyHaveConfigSetup を `Yes` に設定してソリューションをデプロイします。

既存の VPC にデプロイすると、サーチリゾルバースタックのデプロイがタイムアウトする

問題: OpenSearch クラスターにインデックスを作成するためにカスタムリソースをプロビジョニングするネストされたスタックが、次のエラーでタイムアウトします。

```
Embedded stack arn:aws:cloudformation:<region>::stack/<stack-name>-
SearchResolversStack-<ID-string>/<guid> was not successfullycreated: Stack creation
 time exceeded the specified timeout
```

理由: CloudFormation パラメータとして指定されたプライベートサブネットは、S3 にルーティングできません (カスタムリソースは、署名済み URL を使用して実行結果を S3 バケットに書き込む必要があります)。これには、一般的に次の 2 つの理由が考えられます。

1. プライベートサブネットに NAT ゲートウェイが関連付けられていないため、インターネットにアクセスできません。
2. プライベートサブネットが NAT ゲートウェイではなく VPC エンドポイントを使用しており、S3 ゲートウェイエンドポイントが正しく設定されていません。

解決策:

1. ドキュメントに従って、CloudFormation または AWS CLI を使用して、プライベートサブネットで実行されているタスクがインターネットにアクセスできるように、VPC に NAT ゲートウェイをプロビジョニングします。
2. ドキュメントに従って、サブネットのルートテーブルが S3 VPC エンドポイント用に更新されていることを確認します。

アカウントのインポート後にリソースが検出されない

問題: Web UI からアカウントをインポートしましたが、検出プロセスの実行後にリソースが検出されないようです。

理由: 最も可能性の高い理由は次のとおりです。

1. CrossAccountDiscovery CloudFormation パラメータが `SELF_MANAGED` に設定されている場合、グローバルリソース CloudFormation テンプレートはデプロイされていません。

2. CrossAccountDiscovery CloudFormation パラメータが `AWS_ORGANIZATIONS` に設定されている場合、1 つ以上のアカウントが検出されず、[ロールステータス] 列に [デプロイされていません] というエントリがあります。これは、StackSets を使用したグローバルリソーステンプレートの自動デプロイに問題が発生していることを意味します。
3. 検出プロセス ECS タスクのメモリが不足しています。これは、多数のアカウントまたはリソースをインポートするときに発生します。UI の [最後の検出] 列には、DiscoveryTaskFrequency CloudFormation パラメータで指定された値よりも大きい値 (デフォルト値は 15 分) があり、ECS コンソールにメモリ不足エラーが表示されます。

解決策:

1. ドキュメントに従って、グローバルリソーステンプレートを必要なアカウントにデプロイします。
2. AWS でのワークロード検出がデプロイされたリージョンの WdGlobalResources StackSet に移動し、デプロイに失敗したスタックインスタンスのエラーを確認します。
3. Memory CloudFormation パラメータをより大きな値に更新します。最初は 2 倍の値から始め、エラーがなくなるまで値を増やし続けます。

> **Note**
>
> CPU ユニットとメモリ値の特定の組み合わせのみが有効なため、CpuUnits CloudFormation パラメータも更新する必要がある場合があります。組み合わせの完全なリストは、ECS ドキュメントに記載されています。

特定のアカウントでは AWS Config 以外のリソースのみが検出される

問題: このソリューションで検出されるリソースタイプは、サポートされているリソースセクションの表に記載されているもののみです。

理由: この問題の最も一般的な原因は次のとおりです。

1. CrossAccountDiscovery CloudFormation パラメータを `SELF_MANAGED` に設定されている場合に、検出対象の各アカウントのリージョンにリージョンリソースの CloudFormation テンプレートがデプロイされていません。

2. CrossAccountDiscovery CloudFormation パラメータが `SELF_MANAGED` に設定されている場合に、Config が有効になっていない複数のアカウントのリージョンにリージョンリソースの CloudFormation テンプレートがデプロイされているが、CloudFormation パラメータ AlreadyHaveConfigSetup が誤って `Yes` に設定されています。
3. CrossAccountDiscovery CloudFormation パラメータが `AWS_ORGANIZATIONS` に設定されている場合に、検出対象となる各アカウントのリージョンで AWS Config が有効になっていません。`AWS_ORGANIZATIONS` モードでは、組織のポリシーに従って Config を有効にする責任があります。

解決策:

1. ドキュメントに従って、必要なアカウントにリージョンリソーステンプレートをデプロイします。
2. 以前にデプロイしたリージョンリソーススタックを削除し (削除しないと AWS Config が不整合な状態になります)、CloudFormation パラメータ AlreadyHaveConfigSetup を `No` に設定して再デプロイします。
3. 検出対象となる各アカウントのリージョンで AWS Config を有効にします。

AWS サポートに問い合わせる

AWS デベロッパーサポート、AWS ビジネスサポート、または AWS エンタープライズサポートをご利用の場合は、サポートセンターを利用して、このソリューションに関するエキスパートのサポートを受けることができます。次のセクションで、その方法を説明します。

ケースを作成する

1. サポートセンターにサインインします。
2. [ケースを作成] を選択します。

どのようなサポートをご希望ですか?

1. [技術] を選択します。
2. [サービス] で、[ソリューション] を選択します。
3. [カテゴリ] で、[その他のソリューション] を選択します。
4. [重要度] で、ユースケースに最も適したオプションを選択します。

5. [サービス]、[カテゴリ]、[重要度] を入力すると、インターフェイスに一般的なトラブルシューティングの質問へのリンクが表示されます。これらのリンクを使用しても問題を解決できない場合は、[次のステップ: 追加情報] を選択してください。

追加情報

1. [件名] に、質問または問題を要約したテキストを入力します。
2. [説明] に、問題の詳細を入力します。
3. [ファイルを添付] を選択します。
4. AWS サポートがリクエストを処理するために必要な情報を添付します。

ケースの迅速な解決にご協力ください

1. 必要な情報を記入します。
2. [次のステップ: 今すぐ解決またはお問い合わせ] を選択します。

今すぐ解決またはお問い合わせ

1. [今すぐ解決] で解決策を確認します。
2. これらの解決策で問題を解決できない場合は、[お問い合わせ] を選択し、必要な情報を入力して [送信] を選択します。

ソリューションをアンインストールする

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. デプロイ中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 コマンドラインインターフェイスの使用

AWS CLI が自身の環境で使用できるかどうかを調べます。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 が使用可能なことを確認したら、次のコマンドを実行します。

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

デベロッパーガイド

このセクションでは、このソリューションのソースコードと追加のカスタマイズについて説明します。

ソースコード

AWS でのワークロード検出の 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。

デプロイリソースの検索

次の手順に従って、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. このソリューションをデプロイした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、表の上にある検索バーを使用して検索することもできます。

サポート リソース

このソリューションでは、こちら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タイプ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::DynamoDB::Stream	SDK	describeStream
AWS::IAM::AWSManagedPolicy	SDK	getAccountAuthorizationDetails
AWS::ElasticLoadBalancingV2::TargetGroup	SDK	describeTargetGroups
AWS::EC2::Spot	SDK	describeSpotInstanceRequests
AWS::EC2::SpotFleet	SDK	describeSpotFleetRequests

AWS Organizations のアカウント検出モード

AWS でのワークロード検出ソリューションが AWS Organizations にデプロイされると、アカウントの検出はこのソリューションのウェブ UI では管理されなくなります。この場合、アカウントを見つけるために CloudFormation テンプレートのデプロイを管理する必要はありません。

その代わりに、このソリューションでは、AWS Organization 全体に AWS Config アグリゲータを使用して、AWS Config が有効になっている組織内のすべてのアカウントのリソースを検出します。

AWS Config でサポートされていないリソースタイプの場合、このソリューションは AWS CloudFormation StackSets を使用して組織の各アカウントに IAM ロールを自動的にデプロイしま

す。このロールにより、検出プロセスは組織のすべてのアカウントで SDK を呼び出して、これらの補足リソースを検出することができます。

この StackSet は、組織に追加された新しいアカウントにロールを自動的にデプロイし、組織から削除されたすべてのアカウントからロールを削除するように設定されています。

> **Note**
>
> StackSet がスタックインスタンスを管理アカウントにデプロイすることはできません。AWS でのワークロード検出でこのアカウントを検出する場合は、「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」セクションで説明されている標準の AWS CloudFormation のデプロイ方法を使用して、グローバルリソーステンプレートをデプロイする必要があります。

Amazon S3 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. 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 コンソールにサインインします。
3. ロールの名前を [IAM の検索] 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. [アクセス許可ポリシー] で、[管理ポリシー] を展開します。
6. 前の表で説明したアクションがポリシーに含まれていることを確認します。

S3 バケットポリシー

次は、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 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/*"

        }
    ]
  }
```

AWS API

前提条件で説明しているように、このソリューションを既存の VPC にデプロイする場合は、ご使用のプライベートサブネットから次のサービスにアクセスできる必要があります。

API Gateway

- GetAuthorizers
- GetIntegration
- GetMethod
- GetResources
- GetRestApis

Cognito

- DescribeUserPool

構成

- BatchGetAggregateResourceConfig
- DescribeConfigurationAggregators
- ListAggregateDiscoveredResources
- SelectAggregateResourceConfig

DynamoDB Streams

- DescribeStream

Amazon EC2

- DescribeInstances
- DescribeSpotFleetRequests
- DescribeSpotInstanceRequests
- DescribeTransitGatewayAttachments

Amazon Elastic Load Balancer

- DescribeLoadBalancers
- DescribeListeners
- DescribeTargetGroups
- DescribeTargetHealth

Amazon Elastic Kubernetes Service

- DescribeNodeGroup
- ListNodegroups

IAM

- GetAccountAuthorizationDetails
- ListPolicies

Lambda

- GetFunction
- GetFunctionConfiguration
- ListEventSourceMappings

OpenSearch Service

- DescribeDomains
- ListDomainNames

組織

- ListAccounts
- ListAccountsForParent
- ListOrganizationalUnitsForParent
- ListRoots

Amazon Simple Notification Service

- ListSubscriptions

Amazon Security Token Service

- AssumeRole

参照資料

このセクションには、このソリューション固有のメトリクスを収集するためのオプション機能に関する情報、このソリューションに貢献したビルダーのリストが含まれています。

匿名化されたデータの収集

このソリューションには、匿名化された運用メトリクスを AWS に送信するオプションが含まれています。AWS で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。アクティブにすると、次の情報が収集され、AWS に送信されます。

- Solution ID - AWS ソリューション識別子
- 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
- Cost Feature Enabled - ユーザーがコスト機能を使用しているかどうかに関する情報
- Number of Accounts - ユーザーがデプロイ時にオンボーディングしたアカウントの数
- Number of Diagrams - 各デプロイで作成されたダイアグラムの数
- Number of Resources - すべてのオンボーディングしたアカウントで検出されたリソースの数

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、プライバシー通知が適用されます。この機能を無効にするには、AWS CloudFormation テンプレートを起動する前に、次の手順を実施してください。

1. AWS CloudFormation テンプレートをローカルハードドライブにダウンロードします。
2. テキストエディタで AWS CloudFormation テンプレートを開きます。
3. AWS CloudFormation テンプレートのマッピングセクションを次のように変更します。

```
Mappings:
  Solution:
    Metrics:
      CollectAnonymizedUsageMetrics: 'true'
```

変更後:

```
Mappings:
  Solution:
    Metrics:
      CollectAnonymizedUsageMetrics: 'false'
```

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. [テンプレートファイルのアップロード] で、[ファイルの選択] を選択し、ローカルドライブから編集したテンプレートを選択します。
5. [次へ] を選択して、「スタックの起動」の手順に従います。

寄稿者

- Mohsan Jaffery
- Matthew Ball
- Stefano Vozza
- Connor Kirkpatrick
- Chris Deigan
- Nick Lee
- Tim Mekari

リビジョン

公開日: 2020 年 9 月。更新については、GitHub リポジトリにある CHANGELOG.md ファイルを参照してください。

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、本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。本書は、(a) 情報提供のみを目的とし、(b) AWS の現行製品と慣行について説明しており、これらは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上の義務や保証をもたらすものではありません。AWS の製品やサービスは、明示または黙示を問わず、一切の保証、表明、条件なしに「現状のまま」提供されます。お客様に対する AWS の責任は AWS 契約によって規定されています。また、本文書は、AWS とお客様との間の契約に属するものではなく、また、当該契約が本文書によって修正されることもありません。

ソリューションは、Apache ライセンスバージョン 2.0 の条件に基づいてライセンスされています。