aws professional services

AWS Well-Architected Framework のご紹介

目次

“AWS Well-Architected Framework”の概要

“AWS Well-Architected Framework”の活用方法

セキュリティの柱 全体像

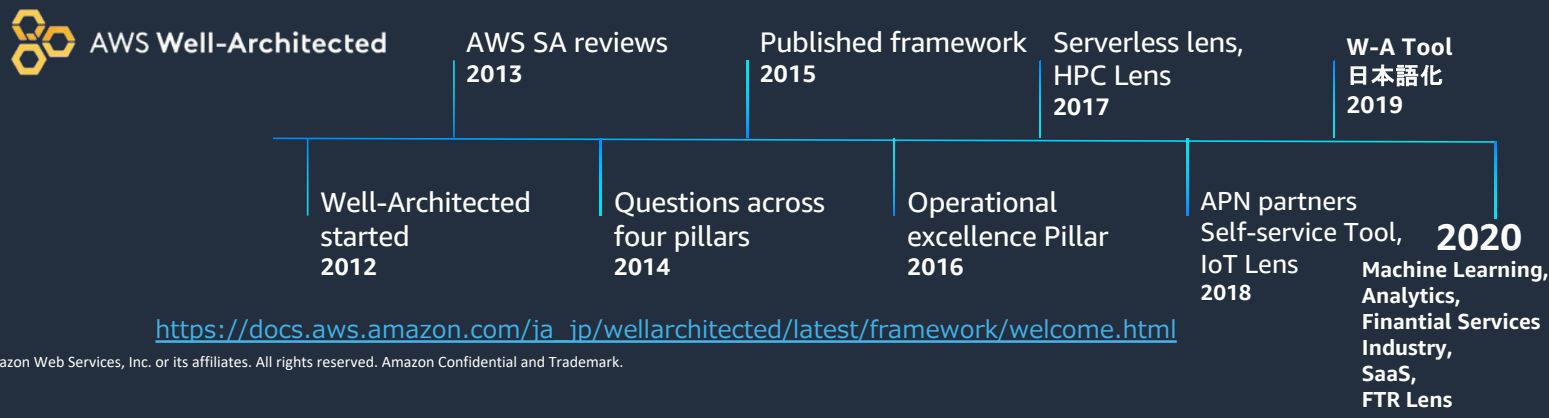
AWS Well-Architected Framework の概要

AWS Well-Architected Framework(W-A) とは?

システム設計・運用の”大局的な”考え方と ベストプラクティス集

・AWS のソリューションアーキテクト (SA)、パートナー様、お客様の 10 年以上にわたる経験から作り上げたもの

・AWS とお客様と共に、W-A も常に進化し続ける



https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/framework/welcome.html

aws professional services

AWS Well-Architected Framework とは？

柱

設計原則
(Design principles)

質問

セキュリティの柱における設計原則

ワークロードを安全に運用する

IDとアクセス管理

検知

インフラストラクチャ保護

データ保護

インシデント対応

アプリケーションのセキュリティ

aws professional services

質問と回答形式でのベストプラクティス(例)

セキュリティ

SEC 2 ユーザIDとマシンIDはどのように管理したらよいでしょうか？

安全に AWS ワークロードを運用するアプローチには、管理する必要があるアイデンティティが 2 種類あります。管理およびアクセス権を付与する必要があるアイデンティティのタイプを理解することで、適切な ID が適切な条件下で適切なリソースにアクセスできるようになります。

- 強力なサインインメカニズムを使用する
- 一時的な認証情報を使用する
- シークレットを安全に保存して使用する
- 一元化されたIDプロバイダーを利用する
- 定期的に認証情報を監査およびローテーションする
- ユーザグループと属性を活用する

柱の分野

質問文

質問のコンテキスト

ベストプラクティス

質問と回答形式でのベストプラクティス(例)

例：セキュリティの質問(抜粋)

[SEC2]　ユーザー ID とマシン ID はどのように管理したらよいでしょうか?

- 強力なサインインメカニズムを使用する
- 一時的な認証情報を使用する
- シークレットを安全に保存して使用する
- 一元化された ID プロバイダーを利用する
- 定期的に認証情報を監査およびローテーションする
- ユーザーグループと属性を活用する

質問と回答形式でのベストプラクティス(例)

例：セキュリティの質問(抜粋)

全項目ベストプラクティスに
則っていないとダメなのか？

- 一時的な認証情報を使用する
- シークレットを安全に保存して使用する
- 一元化された ID プロバイダーを利用する
- 定期的に認証情報を監査およびローテーションする
- ユーザーグループと属性を活用する

質問と回答形式でのベストプラクティス(例)

例：セキュリティの質問(抜粋)

全項目ベストプラクティスに
則っていないとダメなのか？

ベストプラクティスを知った上で、
システム担当者が「(ビジネス的な)判断をする」ための手法
→リスクや改善点の”顕在化”

AWS Well-Architected Framework の活用

AWS Well-Architected Framework の活用

- Well-Architected レビュー
  - AWS Well-Architected フレームワークにおける
    各柱ごとの質問にワークロードを照らし合わせる作業のこと
  - 現在 (As is) のベストプラクティスの
    適用状況を理解し、将来 (To be) のアーキテクチャーを検討する
  - 設計、構築、運用等、ワークロードの特性が
    変わるタイミングで定期的に実施することを推奨

AWS Well-Architected Tool

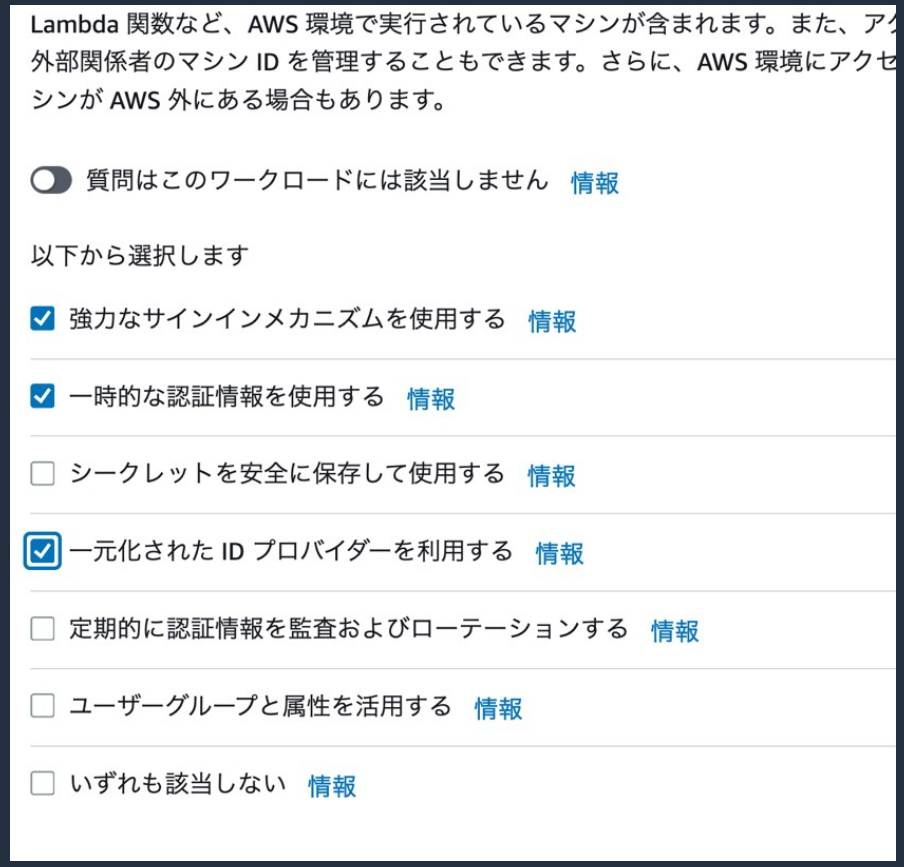
- AWS Well-Architected Tool
  - Well-Architected レビュー時に活用できる AWS の標準サービス
- ★AWS Well-Architected レビューチェックシート
  - Well-Architected レビュー時に活用できるExcelチェックシート(今回用にカスタマイズ作成)
  - 各観点を満たすための設計方式を具体的に確認可能

AWS Well-Architected Tool 使用例

AWS Well-Architected Tool

マネジメントコンソール経由でツールへのアクセスが可能

AWS経験やW-Aの観点に対する知識があれば効率的な利用が可能



レビューチェックシート 使用例

各項目を満たすための具体的な設計方法を詳細に一覧化

W-Aの観点に慣れていない場合でもレビュー観点を具体的に理解可能

※チェックシートは今回Workshop向けに作成しているものとなり、
　カバー範囲は現状セキュリティの柱のみが対象

2.1	強力なサインインメカニズムを使用する	ユーザーがAWS環境にサインイン（ログイン）する方法は適切か？ ・すでに社内で利用されているユーザーディレクトリ (Active Directory等) などがある場合、AWS IAM Identity Center を利用した シングルサインオン (SSO) を利用する。 ・安全なユーザー認証方式を設定する（MFA, パスワードポリシー） ・同じIDを複数のユーザーで共有しない ・ワークロードの利用者（コンシューマー）の認証方式は適切か？
2.2	一時的な認証情報を使用する	長期的な認証情報（アクセスキー）の代わりに、短期的な認証情報（IAMロールやSSO認証）を使用しているか？ ・人間のユーザーがAWSサービスを利用する場合可能な限りSSOによる認証を行う ・マシンからAWSサービスにアクセスする場合、IAMロールにより権限を付与する。 ・コードに認証情報を埋め込まない ・やむをえずにAWSアクセスキーを使う場合、ローテーションを行う

レビュー観点説明資料

特に重要で検討が必要なポイントについては補足説明用の資料を準備

※チェックシートは今回Workshop向けに作成しているものとなり、
カバー範囲は現状セキュリティの柱のみが対象

レビュー観点：セキュリティの基礎/アカウント環境の管理と分離

SEC01-BP01 アカウントを使用してワークロードを分ける

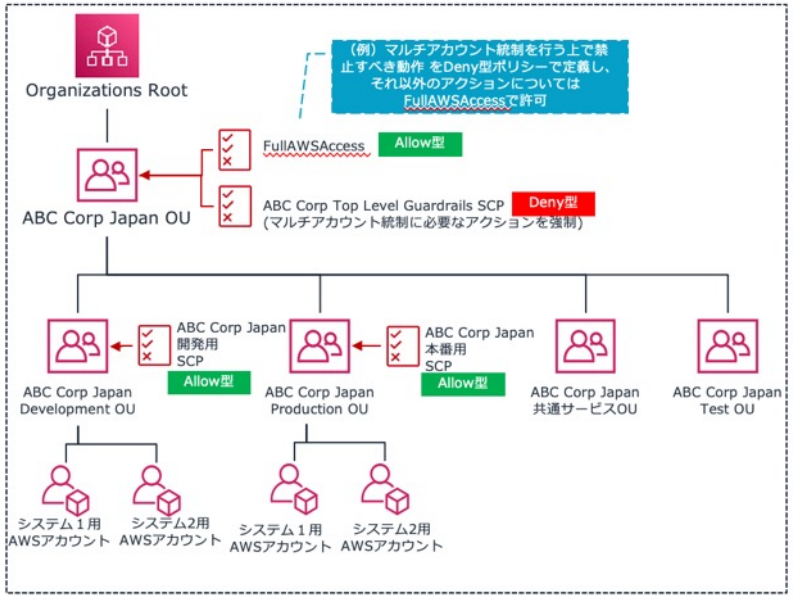
ポイント

- クラウド上で利用される様々なワークロードを一つのAWSアカウント上に配備してしまうと、 アカウント間の干渉や誤操作による問題、、データ漏洩などの原因となる可能性がある。
- ワークロードの種類ごとにAWSアカウントを分けて、適切な統制の仕組みを導入することで、全社的な統制レベルが向上する。

レビュー観点の例

- 複数のアカウント（マルチアカウント）によって、ワークロードや用途（本番、開発、テスト）の環境を分離されているか。
- アカウントに対する統制が行われているか
  - **AWS Organizations**によってアカウントの階層構造を定義し、**SCP (Service Control Policy)** によって各アカウントの権限を制御しているか。（SCPを設定することで、各アカウントのルートユーザーに対しても、AWSの利用を制約することが可能）
  - **AWS Config** や **AWS Security Hub** などの様々なサービスで、複数のアカウントにまたがった管理を実施しているか。
- 新しく作成するアカウントについても正しい設定が行われるか
  - **Landing Zone** を設置し、**AWS Control Tower** により、テンプレートを利用して新しいアカウントを迅速にプロビジョニングすることができます。

アカウント階層構造の例



aws professional services

Well-Architected レビューのタスク

一般的なW-A レビュー(W-A Tool使用時)の実施タスクを一覧化

フェーズ	お客様	AWS or W-A 認定パートナー
レビュー前準備	・対象システムのシステム構成図 ・対象システムのAWSアカウントID ・W-Aレビューリスト(W-A Tool)へのご回答※1 →可能な範囲でレビュー実施までにご回答ください。 ・レビュー当日への参加者のアサイン →システムの基本設計を説明できる方を含めて、可能な限り多くのステークホルダーの方へのご参加されることが望ましいです。またパートナー様もご参加いただけます。	・W-Aレビューリスト(質問集)の事前説明
レビュー当日	・システムのアーキテクチャの説明	・W-Aとレビューの目的の説明 ・アーキテクチャのレビューと説明※1
レビュー後作業	・結果レポートに記載された改善点のプロジェクトへの取り込み	・結果レポートの作成と発行 ・改善点のフォロー

※1 レビューチェックシート/レビュー観点説明資料の併用が可能

aws professional services

Well-Architected レビューのポイント

監査ではない

話し合いであり、重大な問題や改善可能な領域の特定が目的

適切な関係者でディスカッションする

ワークロードの理解と適切な改善を検討

1 回限りのチェックではない

定期的な見直し (KAIZEN)

aws professional services

セキュリティの柱 全体像

Well-Architected Framework セキュリティの柱

カテゴリは11個に分類され、各カテゴリ毎に複数のレビュー項目が存在

No	カテゴリ	項目数	概要
1	セキュリティの基礎	8	アカウント分離やガバナンス・プロセス
2	ID管理	6	認証方式や認証情報の保護
3	アクセス管理	9	アクセス権限の管理
4	検知	4	望ましくない設定変更や不正アクセスの検知
5	ネットワーク保護	4	ネットワーク構成の安全性
6	コンピューティングの保護	6	EC2などの計算資源の保護
7	データ分類	4	データの識別・分類と管理
8	保管中のデータの保護	5	データの暗号化や鍵管理、アクセス管理
9	伝送中のデータの保護	4	証明書、SSL/TLS, IPSec などによる通信の保護
10	インシデント対応	7	インシデントの予測と準備、対応、復旧
11	アプリケーションのセキュリティ	8	設計、開発、デプロイのライフサイクル全体を通じて、アプリケーションのセキュリティ

Well-Architected Framework セキュリティの柱

サンプルとして以下の項目から具体的にいくつかを抜粋し
WA説明資料を用いてご説明(次ページ以降)

No	カテゴリ	項目数	概要
1	セキュリティの基礎	8	アカウント分離やガバナンス・プロセス
2	ID管理	6	認証方式や認証情報の保護
3	アクセス管理	9	アクセス権限の管理
4	検知	4	望ましくない設定変更や不正アクセスの検知
5	ネットワーク保護	4	ネットワーク構成の安全性
6	コンピューティングの保護	6	EC2などの計算資源の保護
7	データ分類	4	データの識別・分類と管理
8	保管中のデータの保護	5	データの暗号化や鍵管理、アクセス管理
9	伝送中のデータの保護	4	証明書、SSL/TLS, IPSec などによる通信の保護
10	インシデント対応	7	インシデントの予測と準備、対応、復旧
11	アプリケーションのセキュリティ	8	設計、開発、デプロイのライフサイクル全体を通じて、アプリケーションのセキュリティ

Well-Architected Framework セキュリティの柱

No	カテゴリ	項目数	概要
1	セキュリティの基礎	8	アカウント分離やガバナンス・プロセス
2	ID管理	6	認証方式や認証情報の保護
3	アクセス管理	9	アクセス権限の管理
4	検知	4	望ましくない設定変更や不正アクセスの検知
5	ネットワーク保護	4	ネットワーク構成の安全性
6	コンピューティングの保護	6	EC2などの計算資源の保護
7	データ分類	4	データの識別・分類と管理
8	保管中のデータの保護	5	データの暗号化や鍵管理、アクセス管理
9	伝送中のデータの保護	4	証明書、SSL/TLS, IPSec などによる通信の保護
10	インシデント対応	7	インシデントの予測と準備、対応、復旧
11	アプリケーションのセキュリティ	8	設計、開発、デプロイのライフサイクル全体を通じて、アプリケーションのセキュリティ

レビュー観点：セキュリティの基礎/アカウント環境の管理と分離

SEC01-BP01 アカウントを使用してワークロードを分ける

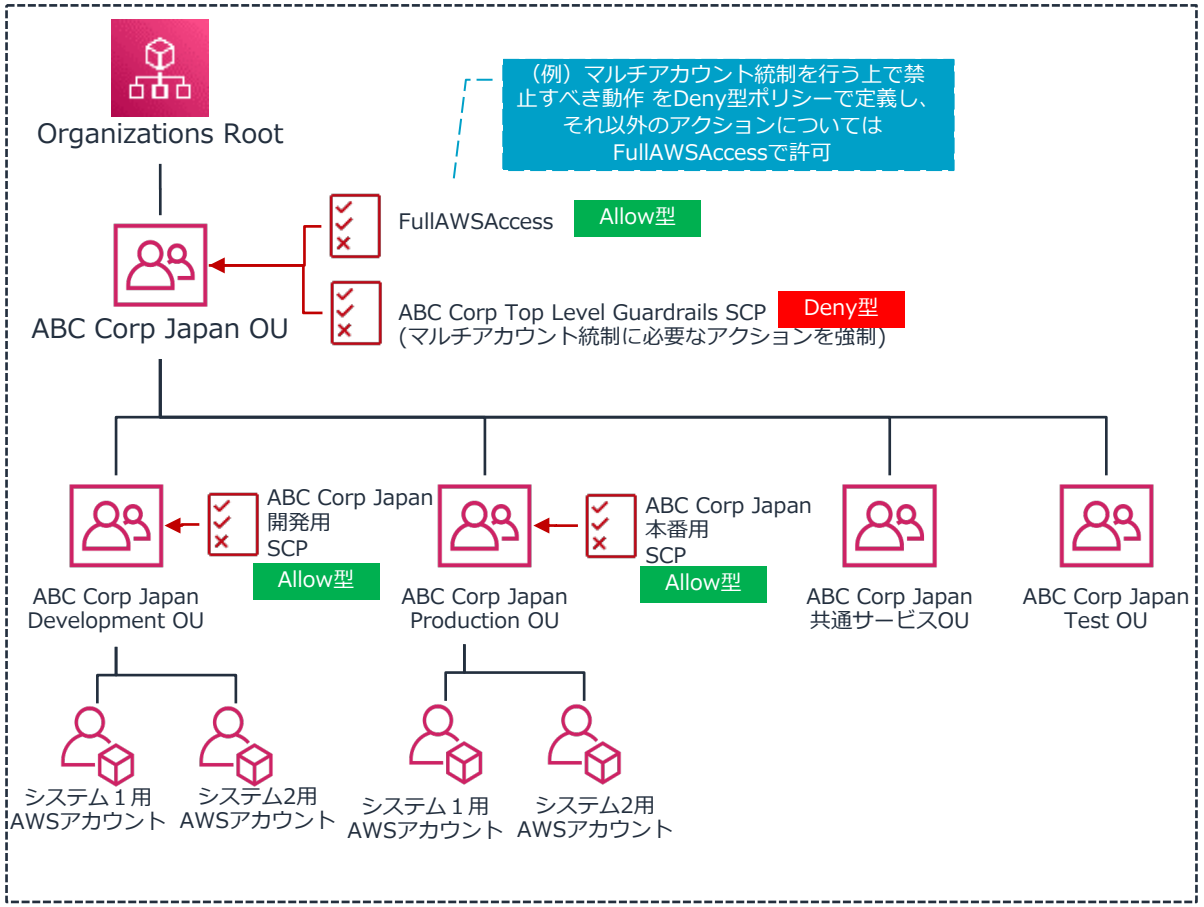
ポイント

- クラウド上で利用される様々なワークロードを一つのAWSアカウント上に配備してしまうと、 アカウント間の干渉や誤操作による問題、、データ漏洩などの原因となる可能性がある。
- ワークロードの種類ごとにAWSアカウントを分けて、適切な統制の仕組みを導入することで、全社的な統制レベルが向上する。

レビュー観点の例

- 複数のアカウント（マルチアカウント）によって、ワークロードや用途（本番、開発、テスト）の環境を分離されているか。
- アカウントに対する統制が行われているか
    - AWS Organizationsによってアカウントの階層構造を定義し、SCP (Service Control Policy) によって各アカウントの権限を制御しているか。（SCPを設定することで、各アカウントのルートユーザーに対しても、AWSの利用を制約することが可能）
    - AWS Config や AWS Security Hub などの様々なサービスで、複数のアカウントにまたがった管理を実施しているか。
- 新しく作成するアカウントについても正しい設定が行われるか
    - Landing Zone を設置し、AWS Control Tower により、テンプレートを利用して新しいアカウントを迅速にプロビジョニングすることができます。

アカウント階層構造の例



aws professional services

参考: AWSアカウントが実現するもの

セキュリティ境界の単位

セキュリティ上の境界

- 他のお客様とのセキュリティの境界の役割を果たす (他のAWSアカウントの構成は見えず、アクセスできない)

リソース管理の単位

リソースの管理単位

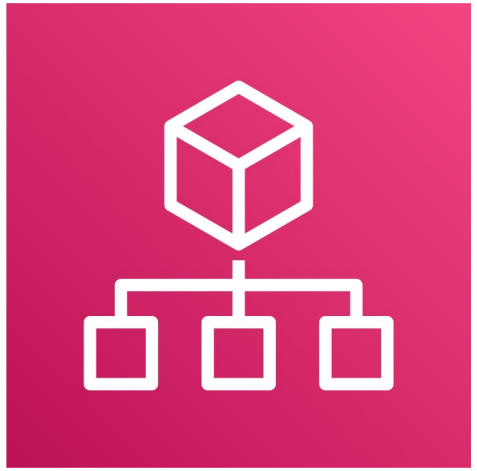
- アカウント単位でAWSのさまざまなリソースを管理する

課金の単位

課金の分離単位

- サービスの利用における課金はAWSアカウントに対して行われる

参考: AWS Organizations概要



AWS Organizations

AWS アカウント全体の一元管理を実現するサービス

サービス概要

- AWSアカウントの作成と管理の機能を提供
  - 対象のAWSアカウントを組織という単位で管理
  - 組織単位(OU)によりAWSアカウントの階層構造を定義
  - Organizations画面からアカウント払い出し可能
  - 支払いの自動での紐づけが可能
  - AWS IAM Identity Center*などのサービス連携による管理性向上
- ポリシーベースでのアカウント管理
  - ポリシーを組織内のアカウントに適用
  - 複数アカウント間でAWSのサービス、リソース、リージョンへのアクセスを管理
- 無償で利用可能

* AWS Single Sign-Onの後継サービス

参考) AWS Organizations のよくある質問
https://aws.amazon.com/jp/organizations/faqs/

参考: AWS Organizationsが実現するもの

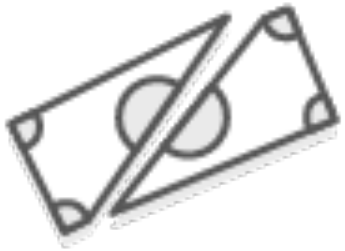
複数のアカウントの一元管理

- 組織単位によるアカウントのグループ化
- ポリシーによるグループ単位での一括統制、管理を実現

アカウント管理の簡素化

- Organizations利用アカウントでのCloudTrailを自動設定
- IAM Identity Center*との連携によるアカウント間シングルサインオンなど

アカウントでの請求の簡素化



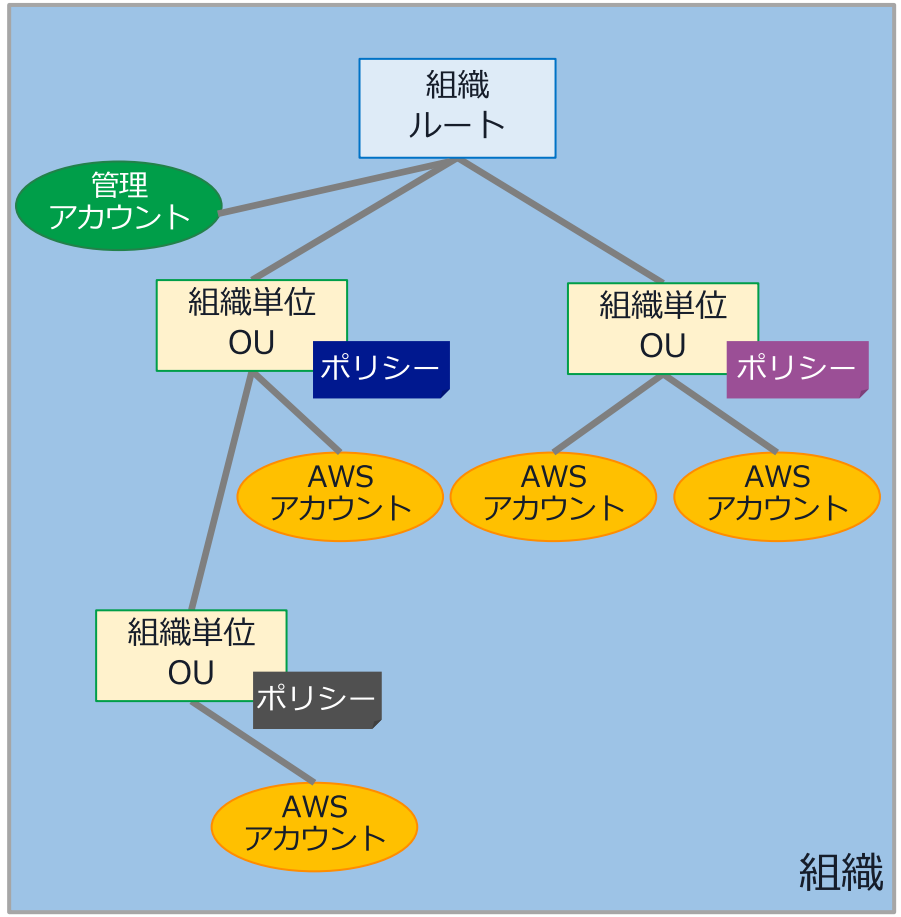
- 複数アカウントの一括請求

* AWS Single Sign-Onの後継サービス

参考: AWS Organizationsの用語

用語	説明
組織	•一元管理可能な複数AWSアカウントの集まり •最低1つの管理アカウント(旧マスターアカウント)から構成される
AWS アカウント	AWS Organizationsで管理する最小単位（メンバーアカウント）
管理 アカウント(※)	アカウントの作成、招待、削除、ポリシーの適用および組織における支払いを行うアカウント
組織単位(OU)	組織内の複数AWSアカウントの論理的なグループ
組織ルート	組織単位 (OU) の階層全体の開始点
サービスコントロールポリシー(SCP)	アカウントに適用するコントロールを定義したドキュメントでAWSサービスのAPIにアクセス可能かを制御（許可・拒否）する

AWS Organizations概念図



参考: AWS Organizations の用語と概念
https://docs.aws.amazon.com/ja_jp/organizations/latest/userguide/orgs_getting-started_concepts.html

Well-Architected Framework セキュリティの柱

No	カテゴリ	項目数	概要
1	セキュリティの基礎	8	アカウント分離やガバナンス・プロセス
2	ID管理	6	認証方式や認証情報の保護
3	アクセス管理	9	アクセス権限の管理
4	検知	4	望ましくない設定変更や不正アクセスの検知
5	ネットワーク保護	4	ネットワーク構成の安全性
6	コンピューティングの保護	6	EC2などの計算資源の保護
7	データ分類	4	データの識別・分類と管理
8	保管中のデータの保護	5	データの暗号化や鍵管理、アクセス管理
9	伝送中のデータの保護	4	証明書、SSL/TLS, IPSec などによる通信の保護
10	インシデント対応	7	インシデントの予測と準備、対応、復旧
11	アプリケーションのセキュリティ	8	設計、開発、デプロイのライフサイクル全体を通じて、アプリケーションのセキュリティ

レビュー観点：IDとアクセス管理 / ID管理
SEC02-BP01 強力なサインインメカニズムを使用する

ポイント

- ユーザーがAWS環境にサインイン（ログイン）する場合の安全な方法を設定するための確認項目。
- AWS環境の利用者（AWSサービスにマネージメントコンソールやAPI, CLIなどでアクセスする）と、ワークロードの利用者（コンシューマ）は異なる認証方式を用いる点に注意。

レビュー観点の例

- すでに社内で利用されてるユーザーディレクトリ (Active Directory等) などがある場合、AWS IAM Identity Center を利用した シングルサインオン (SSO) を利用する。
    - 必要最小限の権限を付与したIAMロールを、AD (Active Directory) グループ等に割り当てる
- 安全なユーザー認証方式を設定する
    - 可能な限りMFAを設定する
    - パスワードポリシーを設定して、推測されやすいパスワードが設定できないようにする
- 同じIDを複数のユーザーで共有しない
- ワークロードの利用者（コンシューマー）については、IAMユーザーではなく、Amazon Cognito user pools を使用し、Cognito user pools または外部のIDプロバイダを使って認証を行う
    - Cognito user pools でも、MFAやパスワードポリシーを適切に設定する
    - IAMユーザーとしてログインする必要があるのは、マネージメントコンソールやAPI/CLIなどでAWSサービスにアクセスする必要がある人のみ。

参考: マルチアカウント環境でのID管理/認証の課題

マルチアカウント環境での課題

- 個別のアカウントでの管理
  - マルチアカウント環境では、アカウント毎にIAMユーザーを作成して管理するのは、高負担＆非効率
  - 管理漏れやパスワードの使いまわしによる危殆化などのリスク

統合的なID管理の仕組みが必要

複数AWSアカウントをIAMユーザーで管理する場合

具体例
ケース１）開発担当の増員(IAMユーザーの増加)
ケース２）新規Bシステムの開発(AWSアカウントの増加)

それぞれのアカウントで
AWS IAMユーザーや
パスワードを管理
しなければならない

ポイント：
IAM”ユーザ”に依存しない、認証の仕組みの構築。

aws professional services

参考: AWS IAM Identity Center とは？

AWS IAM Identity Center

複数AWSアカウントやSAML
ベースアプリの認証を一元管理

サービス概要

- Organizationsの有効化が前提となるサービス
- 複数AWSアカウントおよびアプリへのシングルサインオンを一元管理
  - 内部IDリポジトリ, SAML 2.0 対応外部IDプロバイダー, Active Directoryをユーザープールに利用することができる(いずれか1つで併用不可)
  - 専用のポータルが準備される
  - 多要素認証が可能
  - Control TowerのLanding Zoneに含まれており、適用時にプロビジョニングされる(使用するかは任意)
  - テンプレートに沿ってSAMLアプリを簡単に連携することができる

参考: AWS IAM Identity Center による認証方式 (1)

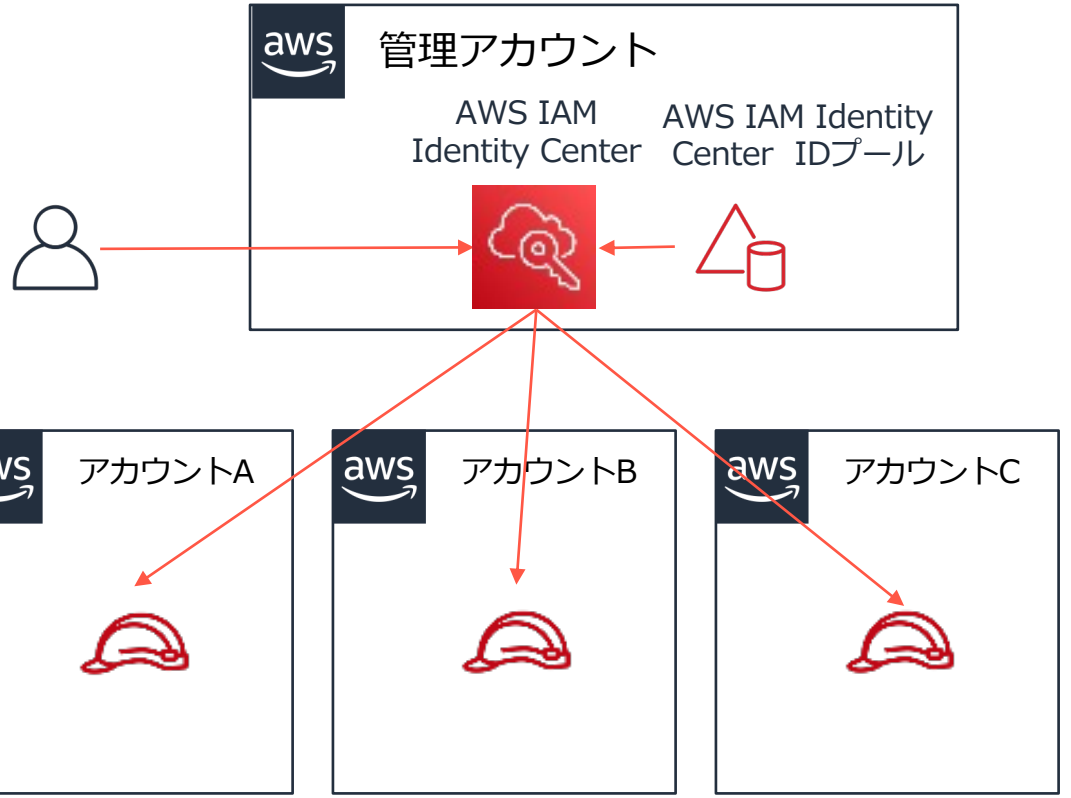
AWS IAM Identity CenterのIDプールを利用した例

方式の説明

- AWS Organizations下のアカウントでは、管理アカウント（支払いアカウント）に設定するAWS IAM Identity Center*を利用してシングルサインオンを構成可能
- 権限は許可セット**またはIAM管理ポリシー、Permissions Boundaryで定義する
- 利用者は、AWS アクセスポータルという専用URLからログインする
- Control Towerを適用した場合構成される(使うのは必須ではない)

* AWS Single Sign-Onの後継サービス
** IAMポリシーと同様のJSONドキュメント

環境アクセスの要件



aws professional services

Well-Architected Framework セキュリティの柱

No	カテゴリ	項目数	概要
1	セキュリティの基礎	8	アカウント分離やガバナンス・プロセス
2	ID管理	6	認証方式や認証情報の保護
3	アクセス管理	9	アクセス権限の管理
4	検知	4	望ましくない設定変更や不正アクセスの検知
5	ネットワーク保護	4	ネットワーク構成の安全性
6	コンピューティングの保護	6	EC2などの計算資源の保護
7	データ分類	4	データの識別・分類と管理
8	保管中のデータの保護	5	データの暗号化や鍵管理、アクセス管理
9	伝送中のデータの保護	4	証明書、SSL/TLS, IPSec などによる通信の保護
10	インシデント対応	7	インシデントの予測と準備、対応、復旧
11	アプリケーションのセキュリティ	8	設計、開発、デプロイのライフサイクル全体を通じて、アプリケーションのセキュリティ

レビュー観点 :検知
SEC04-BP03 イベントへの応答を自動化する

ポイント

- 人為的な労力やミスを軽減するために、自動化の仕組みを取り入れることが重要である。

レビュー観点の例

- セキュリティイベントを検知した際の対応は自動化されているか？
- AWS Config Rules によるシステムのコンプライアンスチェックや自動修復を行なっているか？

実装例

- AWS Config Rules により望ましくないシステム設定を検出して通知、もしくは自動的に修復する。
- AWS GuardDuty により、脅威リスクを検出した際、EventBridgeとSNSを使ってに自動アラートを通知する。
- AWS Step Functions などを使って、インシデントレスポンスをワークフローとして実行する。

参考: CloudTrail で検知したイベント通知の自動化

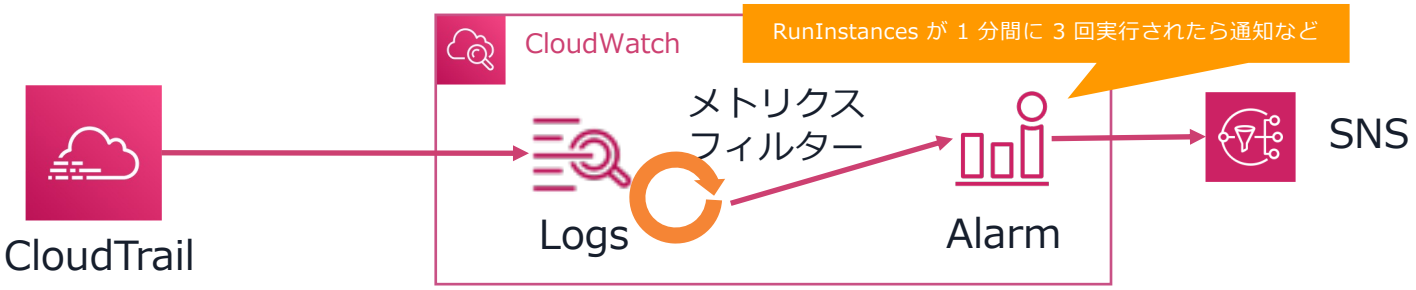
CloudTrail で記録された API 実行の検知方法として下記の 2 つのパターンがあり用途を踏まえて実装要否を検討する
合わせてどこで監視を行うかを決定する (ログやイベントの集約と監視リソースの配置方法など)

1. EventBridge のイベントルールによる監視

ほぼリアルタイムの通知が可能であり、基本的にはこちらを推奨

※イベントでは Get/List/Describe から始まるアクションは、一部を除いて非対応
※ 複雑な判断が必要な場合は Lambda を組み合わせて使用する

2. CloudWatch Logs メトリクスフィルターによる監視



イベントパターンでは表現が難しい場合や、Get/List/Describe への対応が必要な場合に使用

※前提として CloudTrail ログの CloudWatch Logs への出力が必要
※複雑な判断が必要な場合はサブスクリプションフィルタと Lambda を組み合わせて使用する

参考：CloudTrail を使用したモニタリングはどのように実行しますか?
https://docs.aws.amazon.com/ja_jp/awscloudtrail/latest/userguide/cloudtrail-concepts.html#cloudtrail-concepts-monitoring

参考: Amazon GuardDutyの概要

Amazon GuardDuty

悪意のあるアクティビティや不正なアクティビティから保護

※30日間無料使用あり

サービス概要

- セキュリティの観点から脅威リスクを検知するAWSマネージドサービス
- 分析のソースには下記を利用し、メタデータの連続ストリームを分析
  - ✓ VPC Flow Logs
  - ✓ AWS CloudTrail Event Logs
  - ✓ DNS Logs
  - ✓ Kubernetes Audit Logs
- 悪意のあるIPアドレス、異常検出、機械学習などの統合脅威インテリジェンスを使用して脅威を認識
- Malware保護機能によりEC2インスタンスまたはコンテナワークロードの疑わしい動作を検出
- 結果は自動的に EventBridgeに送信
  - ✓ S3バケットにエクスポートすることも可能
  - ✓ 出力先のS3バケットは同じアカウントでも、他のアカウントでも対応可

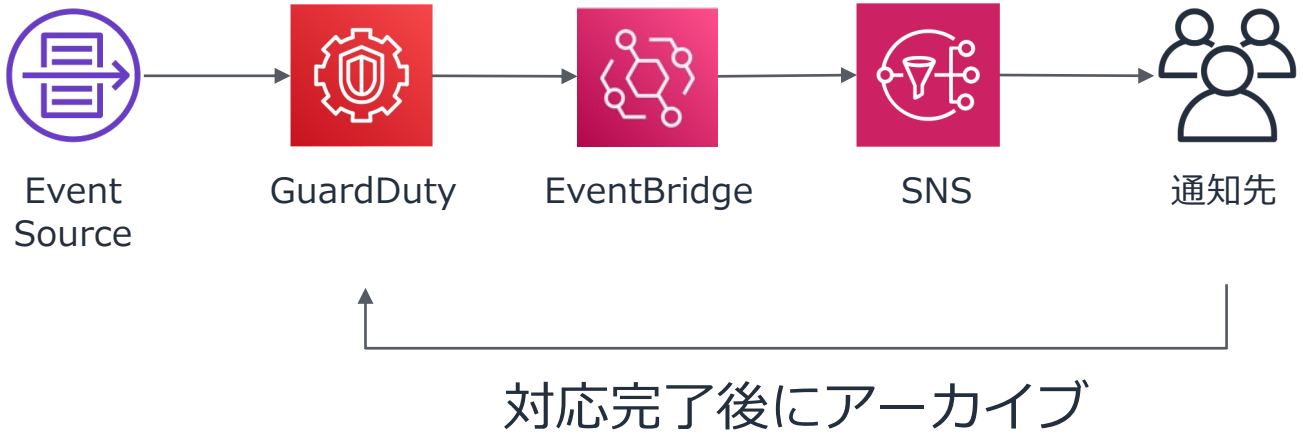
参考:
https://aws.amazon.com/jp/guardduty/
Amazon GuardDuty
https://aws.amazon.com/jp/guardduty/faqs/
Amazon GuardDuty のよくある質問
https://d1.awsstatic.com/webinars/jp/pdf/services/20180509_AWS-BlackBelt_Amazon-GuardDuty.pdf
[AWS Black Belt Online Seminar] Amazon GuardDuty

参考： Amazon GuardDutyの通知

結果通知の方法

- サービス間連携
  - EventBridgeと連携し、通知の構成が可能
  - 通知は各リージョンごとに設定が必要(同一リージョン)
  - SNSや通知機能と連携し、オペレータに対する通知が可能
  - EventBridgeを使用して対象重要度を絞った通知も可能※
- アーカイブ
  - 対応完了後はアーカイブすることで非表示にできる

連携イメージ



※参考:
Amazon EventBridge でのコンテンツのフィルタリング導入開始
https://aws.amazon.com/jp/about-aws/whats-new/2020/02/introducing-content-filtering-amazon-eventbridge/

参考: Amazon GuardDuty検知への対応方針のパタン

検知方法

- GuardDutyはニアリアルにログを解析する
- 重要度高のFindingsは脅威が差し迫っていることを通知するものが多い
- 結果の保管期間は90日(引き上げは不可)
- 結果を検知する仕組みの構築が必要

運用のパターン(以下から選択)

パターン①：定期実行
- 定期的にコンソールから結果を確認
- 確認の結果、必要なアクションを実施
- 対応完了後アーカイブ

パターン②：監視による常時検出
- FindingsのHighやMiddleを監視
- SNS等の通知を受けて必要なアクションを実施
- 対応完了後にアーカイブ

パターン③：監視による常時検出＋自動対応
- 対応方法を事前に定義(インスタンスの隔離、バケットの閉塞など)
- EventBridgeから自動対処
- 自動アーカイブし、結果を通知

Well-Architected Framework セキュリティの柱

その他の項目についての詳細は「WA説明資料」または「AWS公式ドキュメント」を参照

No	カテゴリ	項目数	概要
1	セキュリティの基礎	8	アカウント分離やガバナンス・プロセス
2	ID管理	6	認証方式や認証情報の保護
3	アクセス管理	9	アクセス権限の管理
4	検知	4	望ましくない設定変更や不正アクセスの検知
5	ネットワーク保護	4	ネットワーク構成の安全性
6	コンピューティングの保護	6	EC2などの計算資源の保護
7	データ分類	4	データの識別・分類と管理
8	保管中のデータの保護	5	データの暗号化や鍵管理、アクセス管理
9	伝送中のデータの保護	4	証明書、SSL/TLS, IPSec などによる通信の保護
10	インシデント対応	7	インシデントの予測と準備、対応、復旧
11	アプリケーションのセキュリティ	8	設計、開発、デプロイのライフサイクル全体を通じて、アプリケーションのセキュリティ

professional
services

Thank you!