[AWS Black Belt Online Seminar]

AWS Identity and Access Management (AWS IAM) ~ベストプラクティスで学ぶAWSの認証・認可~ Part1

Solutions Architect 保坂 匠
2019/1/29

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive

自己紹介

保坂 匠 (ほさか たくみ)

アマゾン ウェブ サービス ジャパン
ソリューション アーキテクト

普段の業務

金融機関のお客様のクラウドへのマイグレーション支援

好きなAWSサービス

AWS Identity and Access Management (IAM)

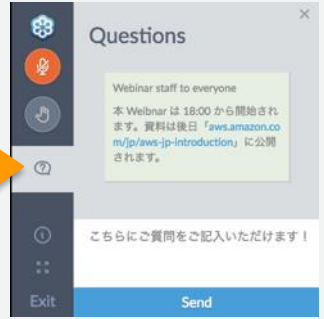
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
  お答えできませんのでご了承下さい

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2019年1月29日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(http://aws.amazon.com)にてご確認ください。

- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 価格は税抜表記となっています。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。

- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at http://aws.amazon.com/agreement/. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

AWS IAMのベストプラクティス

IDと認証情報の管理	✓ AWSアカウントのルートユーザーアクセスキーをロックする ✓ 個々のIAMユーザーを作成 ✓ ユーザーの強力なパスワードポリシーを設定 ✓ アクセスキーを共有しない ✓ 特権ユーザーに対してMFAを有効化する
アクセス権限の管理	✓ AWS管理ポリシーを使用したアクセス許可の使用開始 ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する ✓ 追加セキュリティに対するポリシー条件を使用する ✓ 最小権限を付与する ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する
権限の委任	✓ Amazon EC2インスタンスで実行するアプリケーションに対し、ロールを使用する ✓ ロールを使用したアクセス許可の委任
IDと権限のライフサイクル管理	✓ AWSアカウントのアクティビティの監視 ✓ アクセスレベルを使用して、IAM権限を確認する ✓ 不要な認証情報を削除する ✓ 認証情報を定期的にローテーションする

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/best-practices.html

AWS IAMのベストプラクティス

IDと認証情報の管理	✓ AWSアカウントのルートユーザーアクセスキーをロックする ✓ 個々のIAMユーザーを作成 ✓ ユーザーの強力なパスワードポリシーを設定 ✓ アクセスキーを共有しない ✓ 特権ユーザーに対してMFAを有効化する
アクセス権限の管理	✓ AWS管理ポリシーを使用したアクセス許可の使用開始 ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する ✓ 追加セキュリティに対するコンディション条件を使用する ✓ 最小権限を付与する ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する
権限の委任	✓ Amazon EC2インスタンスで実行するアプリケーションに対し、ロールを使用する ✓ ロールを使用したアクセス許可の委任
IDと権限のライフサイクル管理	✓ AWSアカウントのアクティビティの監視 ✓ アクセスレベルを使用して、IAM権限を確認する ✓ 不要な認証情報を削除する ✓ 認証情報を定期的にローテーションする

Part 1
(本日お話する範囲)

Part 2

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/best-practices.html

本日 (Part1) のアジェンダ

- AWS IAMの概要
- IDと認証情報の管理
- アクセス権限の管理
- まとめ

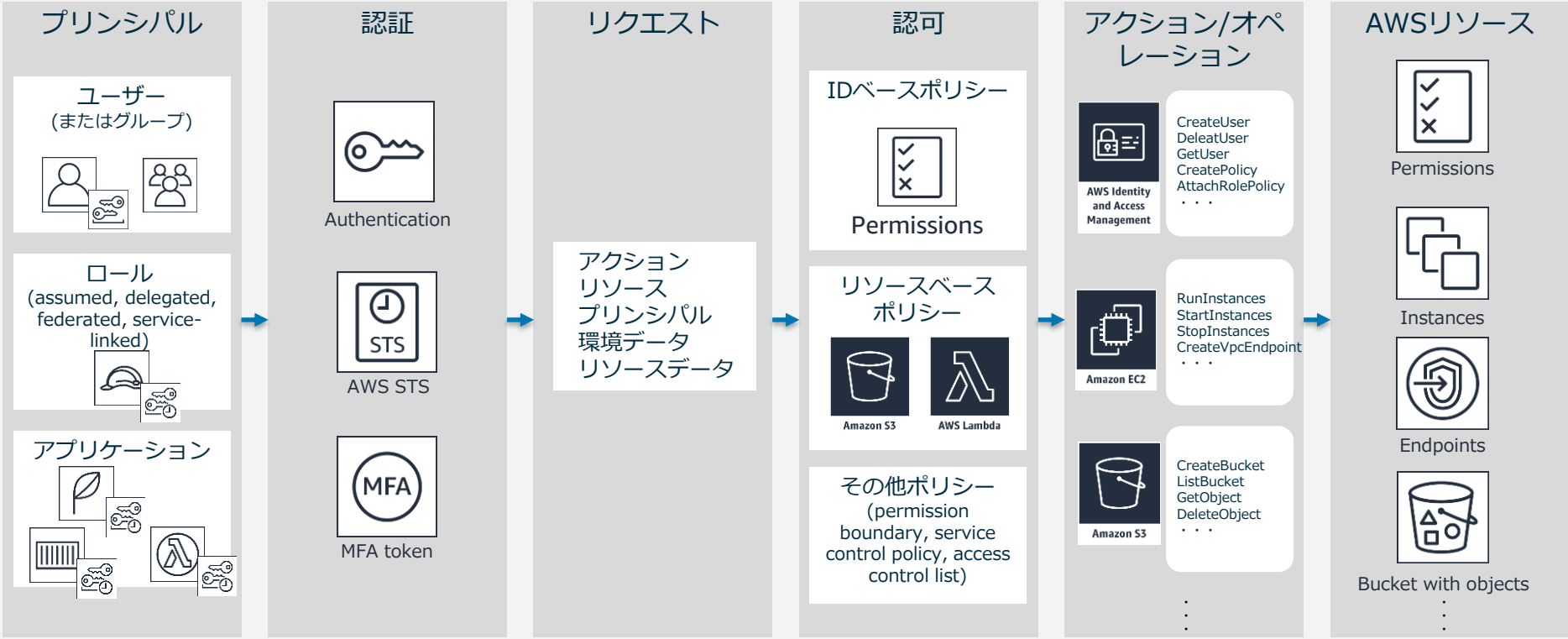
AWS IAMの概要

AWS Identity and Access Management (IAM)とは

- AWSリソースをセキュアに操作するために、認証・認可の仕組みを提供するマネージドサービス

- 各AWSリソースに対して別々のアクセス権限をユーザー毎に付与できる
- 多要素認証(Multi-Factor Authentication : MFA)によるセキュリティの強化
- 一時的な認証トークンを用いた権限の委任
- 他のIDプロバイダーで認証されたユーザーにAWSリソースへの一時的なアクセス
- 世界中のAWSリージョンで同じアイデンティティと権限を利用可能
  - データ変更は結果整合性を保ちながら全リージョンに伝搬
- AWS IAM自体の利用は無料

AWSリソースにアクセスするしくみ



IDと認証情報の管理

- ✓ AWSアカウントのルートユーザーアクセスキーをロックする
- ✓ 個々のIAMユーザーを作成
- ✓ ユーザーの強力なパスワードポリシーを設定
- ✓ アクセスキーを共有しない
- ✓ 特権ユーザーに対してMFAを有効化する

AWSアカウントのルートユーザー

- そのアカウントの全てのAWSサービスとAWSリソース全てに完全なアクセス権を持つユーザー

- AWSマネジメントコンソールへはAWSアカウントを作成したときのメールアドレス/パスワードでサインイン
- IAMで設定するアクセスポリシーではアクセス許可を制限できない
  - AWS Organizationsのサービスコントロールポリシー(SCP)によってサービスを制限可能
- 極力ルートユーザーを使用しないでください！
  - とはいえルートユーザーでの認証が必要なタスクもある

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_root-user.html

ルートユーザーの認証が必要なAWSタスクの例

- ルートユーザーのメールアドレスやパスワードの変更
- IAMユーザーによる課金情報へのアクセスのActivate/Deactivate
- 支払オプションの変更
- AWSサポートプランの変更
- IAMユーザーへのアクセス許可のリストア
- 無効な制約を設定したAmazon S3 バケットポリシーの修正
- 脆弱性診断フォームの提出
- 逆引きDNS申請
- CloudFrontキーペアの作成
- AWSアカウントの解約

(2019年1月29日時点)

https://docs.aws.amazon.com/ja_jp/general/latest/gr/aws_tasks-that-require-root.html

アクセスキー

- AWSアカウントのルートユーザーまたはIAMユーザーの長期的な認証情報
  - 手動で取り消すまで有効

- アクセスキーを用いてAWS CLIやAWS SDK等からリクエストに署名
- アクセスキーID/シークレットアクセスキーで構成される
- 安全なローテーションのために、最大2つのアクセスキーを持つことができる

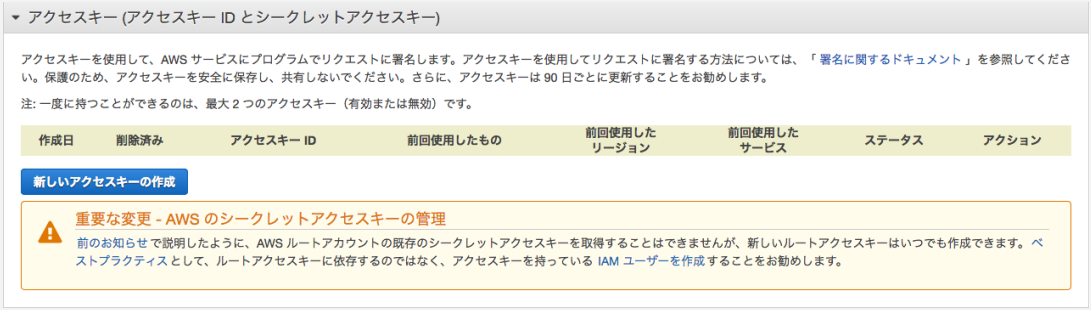
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_credentials_access-keys.html

✓ AWSアカウントのルートユーザーアクセスキーをロックする

Lock Away Your AWS Account Root User Access Keys

- ルートユーザーのアクセスキーは削除してください！

- すでに持っている場合は削除してください！
  - ルートユーザーでサインインし、セキュリティ認証情報のページからアクセスキーを削除
- ルートユーザーの認証情報を他者に開示したり、プログラムに埋め込んだりしないでください！



IDと認証情報の管理

- ✓ AWSアカウントのルートユーザーアクセスキーをロックする
- ✓個々のIAMユーザーを作成
- ✓ ユーザーの強力なパスワードポリシーを設定
- ✓ アクセスキーを共有しない
- ✓ 特権ユーザーに対してMFAを有効化する

IAMユーザー

- AWSで作成するエンティティ (ユーザーまたはアプリケーション)

- 名前と認証情報で構成される
- IAMユーザーを識別する方法
  - ユーザーの「フレンドリ名」：ユーザー作成時に指定
    “Alice”と“alice”は同一のユーザーと見なされ、作成しようとするとエラー
  - ユーザーのARN (Amazon Resource Name)：リソースポリシーのPrincipal要素で指定
    例：arn:aws:iam::*0123456789012*:user/*Alice*
  - ユーザーの一意の識別子：フレンドリ名を再利用したいとき等に権限のエスカレーションを避けることができる
    例：AIDAJQABLZS4A3QDU576Q
- 認証情報
  - コンソールパスワード
  - アクセスキー

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_users.html

✓ 個々のIAMユーザーの作成

Create Individual IAM Users

- 個別のIAMユーザーを作成してください。

- 必要な場合を除き、AWSアカウントのルートユーザー認証情報を使用してAWSにアクセスしないでください！
- 個別のIAMユーザーを作成するメリット
  - 認証情報を個別に変更(ローテーション)できる
  - アクセス許可をいつでも変更、無効化できる
  - Amazon CloudTrailログからアクションを追跡できる

IDと認証情報の管理

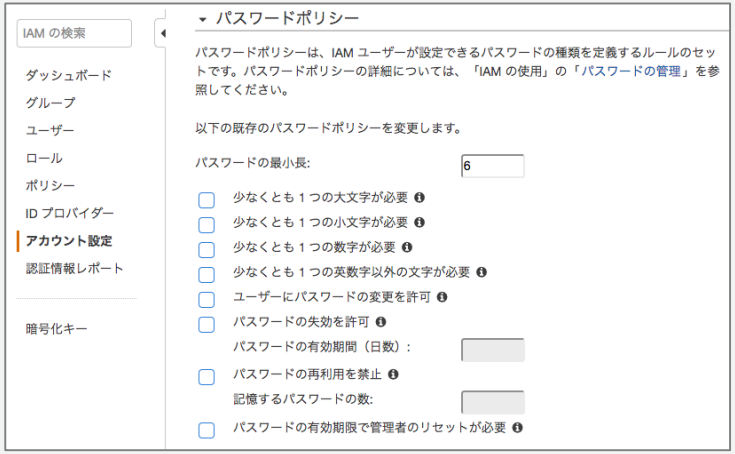
- ✓ AWSアカウントのルートユーザーアクセスキーをロックする
- ✓ 個々のIAMユーザーを作成
- ✓ ユーザーの強力なパスワードポリシーを設定
- ✓ アクセスキーを共有しない
- ✓ 特権ユーザーに対してMFAを有効化する

✓ ユーザーの強力なパスワードポリシーを設定

Configure a Strong Password Policy for Your Users

- 強力なパスワードポリシーを設定してください！

- パスワードに要求される強度とパスワード管理のポリシーを設定可能
  - ✓ 最小文字数
  - ✓ 少なくとも1つの英大文字
  - ✓ 少なくとも1つの英小文字
  - ✓ 少なくとも1つの数字
  - ✓ 少なくとも1つの特殊文字
  - ✓ ユーザー自身によるパスワード変更の許可
  - ✓ パスワードの有効期限
  - ✓ パスワード再利用禁止の世代数
  - ✓ 管理者による期限切れパスワードのリセット



- AWSアカウントのルートユーザーのパスワードポリシーには適用されない

IDと認証情報の管理

- ✓ AWSアカウントのルートユーザーアクセスキーをロックする
- ✓ 個々のIAMユーザーを作成
- ✓ ユーザーの強力なパスワードポリシーを設定
- ✓ アクセスキーを共有しない
- ✓ 特権ユーザーに対してMFAを有効化する

✓ アクセスキーを共有しない

Do Not Share Access Keys

- 複数の人がAWSリソースへのアクセス権を共有したい場合でも、アクセスキーを共有しないでください！
- AWS へのアクセスを必要とするアプリケーションの場合は、IAM ロールを使用して一時的セキュリティ認証情報を取得する (Part2で解説)

- 情報の置き場に注意
  - GitHubリポジトリ
  - AMIの中への埋め込み
  - 設計書等のドキュメント内に記載
  - プレーンテキストでの保管
  - ハードコーディング(AWS認証情報ファイル/環境変数)

IDと認証情報の管理

- ✓ AWSアカウントのルートユーザーアクセスキーをロックする
- ✓ 個々のIAMユーザーを作成
- ✓ ユーザーの強力なパスワードポリシーを設定
- ✓ アクセスキーを共有しない
- ✓特権ユーザーに対してMFAを有効化する

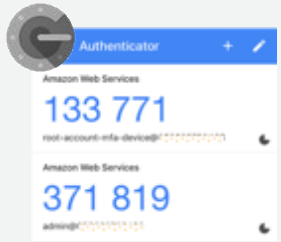
MFA (Multi-Factor Authentication:多要素認証)

- パスワードやアクセスキーによる認証に追加して、セキュリティを強化するしくみ

- AWSがサポートするMFAメカニズム:
  - 仮想MFAデバイス
  - U2Fセキュリティキー
  - ハードウェアMFAデバイス
- ルートユーザー、IAMユーザーの各IDに個別のMFA設定が可能
- MFA条件を指定したポリシーを関連付けできる対象:
  - IAMユーザーまたはIAMグループ
  - Amazon S3バケット、Amazon SQSキュー、Amazon SNSトピック等のリソース
  - IAMロールの信頼ポリシー

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_credentials_mfa.html

AWSがサポートするMFAメカニズム



	ソフトウェア	New! ハードウェア	
製品	Google Authenticator, Authy 2-Factor Authentication	Yubikeyセキュリティキー	Gemalto
形式	スマホアプリ	USBスティック型	トークン型
コスト	無料	有料(4,500円程度)	有料(2,000円程度)
機能	単一のデバイスで複数のトークンをサポート	単一のセキュリティキーで複数のルートユーザー/IAMユーザーをサポート 手入力不要	不正開封防止用キーホルダー型デバイスで内蔵電池により単体でOTPを発行

https://aws.amazon.com/jp/iam/details/mfa/

✓ 特権ユーザーに対してMFAを有効化する

Enable MFA for Privileged Users

- AWSアカウントのルートユーザーや強い権限を持つIAMユーザーにはMFAを有効化し、通常利用しないようにしてください！
  - MFAデバイスも厳重に管理してください！

- 認証プロセスを完了するには、ユーザーの認証情報とデバイス生成のレスポンスが必要になるため、アイデンティティの保護に役立つ
- MFAデバイスの紛失/盗難/不具合が発生したら、代替の認証要素を使って認証し、新しいMFAデバイスを有効化し、パスワードも変更する

https://aws.amazon.com/jp/iam/faqs/#MFA_FAQs

ここまでのまとめ：IDと認証情報の管理に関するベストプラクティス

- ✓ AWSアカウントのルートユーザーアクセスキーをロックする
- ✓ 個々のIAMユーザーを作成
- ✓ ユーザーの強力なパスワードポリシーを設定
- ✓ アクセスキーを共有しない
- ✓ 特権ユーザーに対してMFAを有効化する

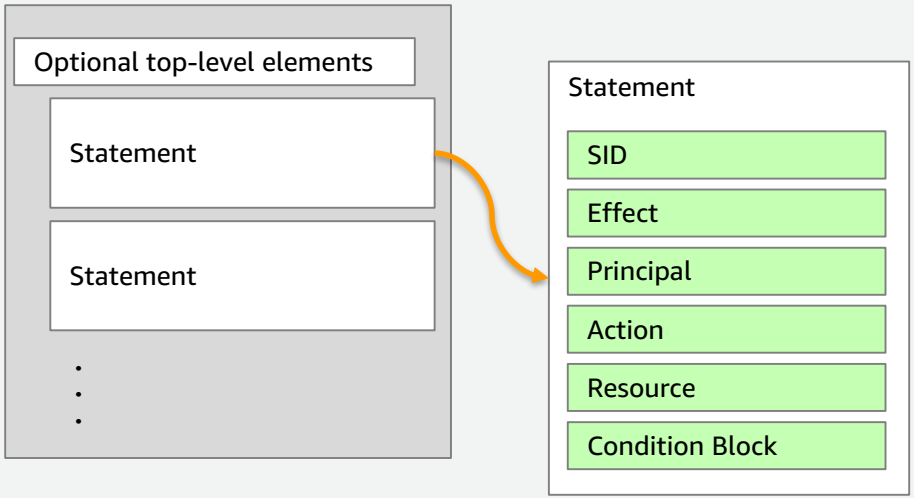
アクセス権限の管理

- ✓ AWS管理ポリシーを使用したアクセス許可の使用開始
- ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する
- ✓ 追加セキュリティに対するポリシー条件を使用する
- ✓ 最小権限を付与する
- ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する

ポリシー

- IAMアイデンティティやAWSリソースに関連づけることによってアクセス許可を定義することができるオブジェクト

- 通常、JSONポリシードキュメントでアクセス条件を記述
- ポリシードキュメントは1つ以上のStatementブロックで構成



https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_grammar.html

AWSがサポートするポリシータイプ

- アイデンティティベースのポリシー
  - 管理ポリシー
    - AWS管理ポリシー
    - カスタマー管理ポリシー
  - インラインポリシー
- リソースベースのポリシー
  - AWS IAMロールの信頼ポリシー、Amazon S3のバケットポリシー、Amazon SNSトピックのアクセス許可、Amazon SQSキューのアクセス許可
- パーミッションバウンダリー
  - AWS IAMアクセス許可の境界、AWS Organizationsサービスコントロールポリシー (SCP)
- アクセスコントロールポリシー (ACL)
  - Amazon S3のバケットのACL、Amazon VPCのサブネットのACL
- セッションポリシー

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies.html

アイデンティティベースのポリシー

- 管理ポリシー
  - 複数のIAMユーザー、IAMグループ、IAMロールに関連付け可能 (最大10個)
  - 再利用可能
  - 一元化された変更管理
  - バージョニングとロールバック
  - 種類
    - AWS管理ポリシー
    - カスタム管理ポリシー

- インラインポリシー
  - 単一のIAMユーザー、IAMグループ、IAMロールに直接埋め込む

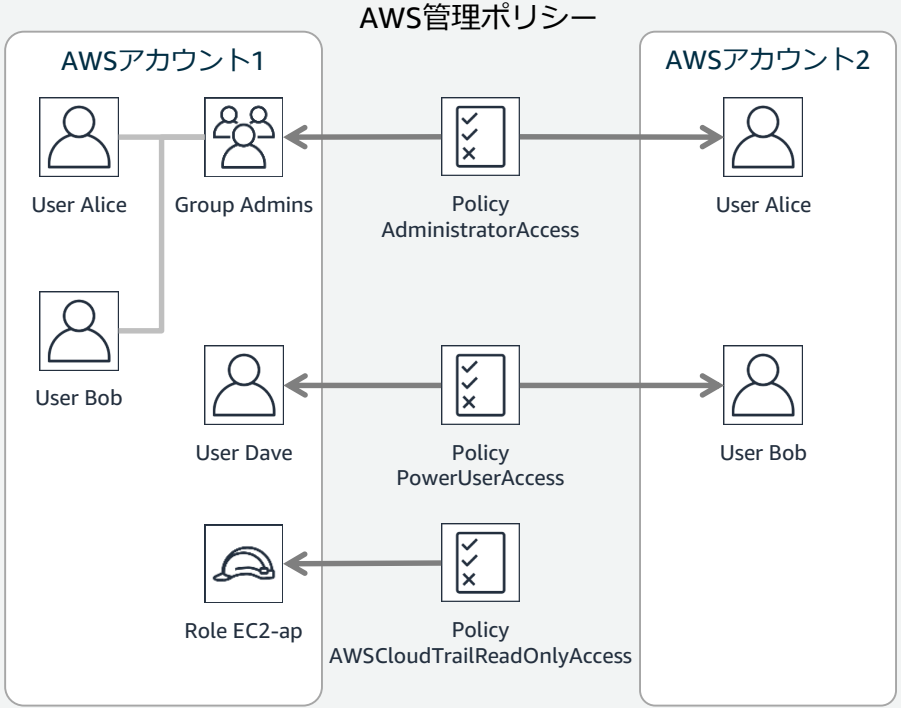
AWS管理ポリシー

- AWSにより事前定義された管理ポリシー

- AWSが作成および管理され、編集不可
- すべてのAWSアカウントで利用可能
  - AWSによる管理

    例：AmazonEC2FullAccess
    AmazonS3ReadOnlyAccess
  - ジョブ機能

    例：AdministratorAccess
    SecurityAudit
    DataScientist
- AWSにより更新される



✓ AWS管理ポリシーを使用したアクセス許可の使用開始

Get Started Using Permissions With AWS Managed Policies

- AWS管理ポリシーを用いると多くのユースケースですぐにポリシーの適用を開始できる

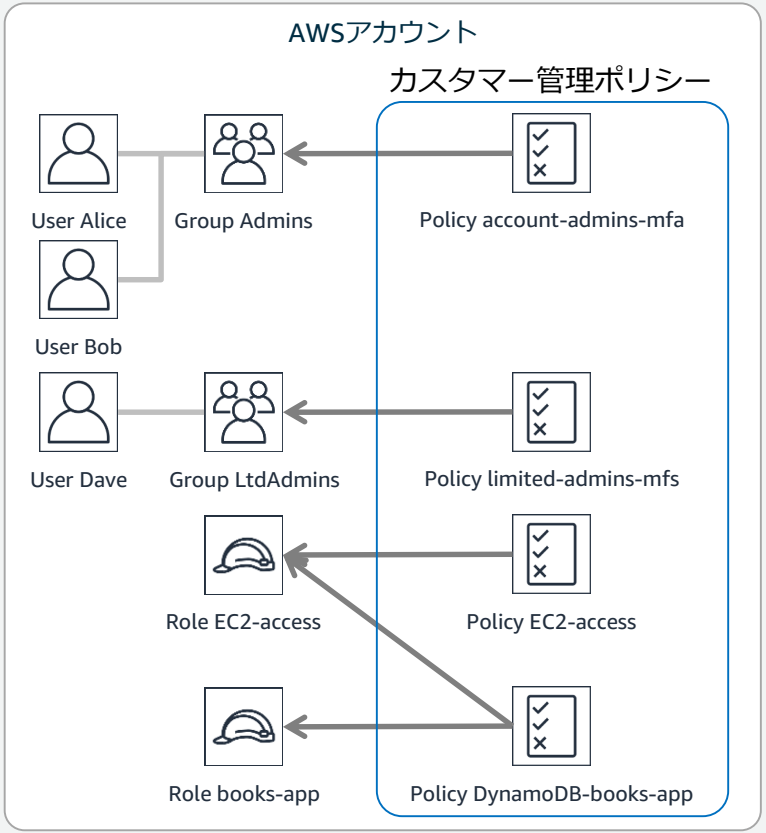
- 適切なアクセス権限を付与するにはIAMポリシーの詳細な知識が必要
- まずはポリシードキュメントの扱いに慣れる

アクセス権限の管理

- ✓ AWS管理ポリシーを使用したアクセス許可の使用開始
- ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する
- ✓ 追加セキュリティに対するポリシー条件を使用する
- ✓ 最小権限を付与する
- ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する

カスタマー管理ポリシー

- AWSアカウントで管理することができるカスタムポリシー
- AWS管理ポリシーでは要件を満たせない場合等にカスタマー管理ポリシーを適用



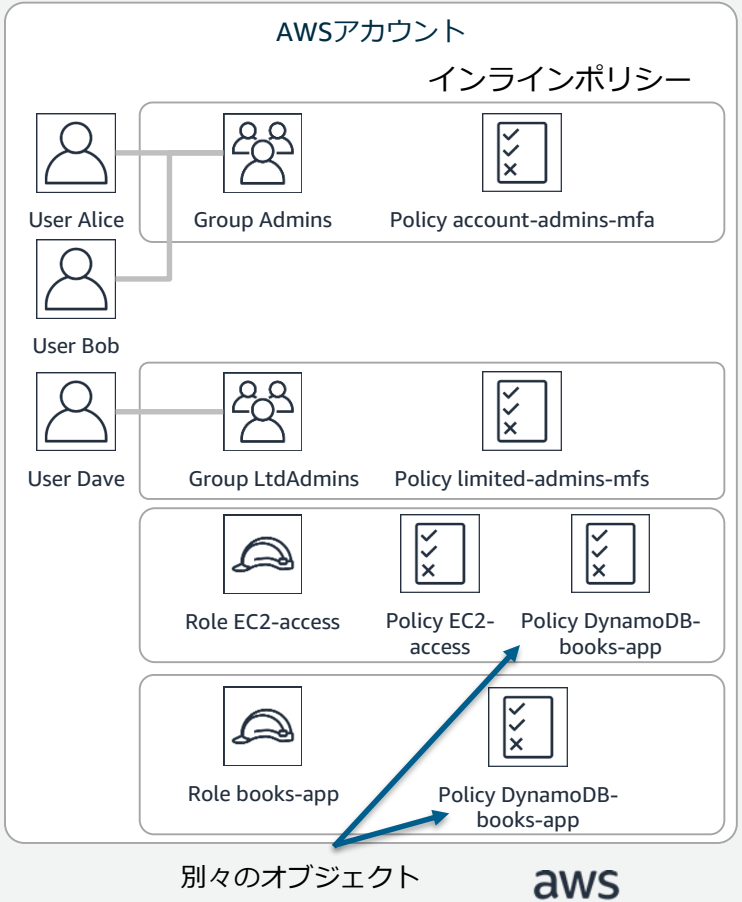
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#customer-managed-policies

インラインポリシー

- 1つのIAMエンティティ (IAMユーザー、IAMグループ、IAMロール)に直接埋め込まれるポリシー

- IAMエンティティに紐づいた固有のオブジェクト

  例えば、右図のEC2-accessロールとbooks-appロールのインラインポリシーDynamoDB-books-appは別物のポリシーオブジェクト
- IAMエンティティを削除するとインラインポリシーも削除される
- IAMエンティティとポリシーとの厳密な1対1の関係を維持する必要がある場合等にインラインポリシーを適用



https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#inline-policies

IAM Best Practice !

✓ インラインポリシーではなくカスタマー管理ポリシーを使用する Use Customer Managed Policies Instead of Inline Policies

- カスタマー管理ポリシーはカスタマイズ可能で、再利用性も高く管理面で有利

- カスタマー管理ポリシーの利点は全ての管理ポリシーを1ヶ所で確認できること
- インラインポリシーの利用はできるだけ避けてください。
- インラインポリシーはカスタマー管理ポリシーに変換することが可能

アクセス権限の管理

- ✓ AWS管理ポリシーを使用したアクセス許可の使用開始
- ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する
- ✓追加セキュリティに対するポリシー条件を使用する
- ✓ 最小権限を付与する
- ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する

ポリシードキュメントの主な要素

要素	概要
Version	ポリシー言語のバージョン。”2012-10-17”が現行バージョン。Version要素を含めないとポリシー変数(${aws:username}等)は文字列として扱われる。
Statement	アクセス許可に関する複数要素 (Effect/Action/Resource等) を含むステートメントブロック。複数のステートメントブロックを並べることができる。
Effect	“Allow”または”Deny”。ステートメントの結果を許可または明示的な拒否にするか指定する。
Principal	リソースベースのポリシー (バケットポリシーや信頼ポリシー等) に記述する。リソースへのアクセスを許可または拒否するIAMエンティティ (IAMユーザー、フェデレーテッドユーザー、IAMロール、AWSサービス等) をARN形式で指定する。
Action	Effect要素で許可または拒否する対象となる特定のアクションを記述する。大文字小文字の区別はされない。各AWSサービスを識別する名前空間 (iam, ec2, s3等) でサポートされるアクションが定義されている。
Resource	Action要素の対象となる特定のリソースをARN形式で記述する。指定したAction (ec2:DescribeInstances等) によっては個々のリソースを指定することができず、”* (ワイルドカード)”を指定する必要がある。
Condition	ポリシーを実行する条件を指定することができる。Condition要素は条件演算子、ポリシー変数、条件値から構成される。

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements.html

Principal要素

- AWSリソースへのアクセスが許可/拒否されるIAMエンティティを指定する

- リソースベースポリシーで使用
- AWSアカウント、IAMユーザー、IAMロール、フェデレーティッドユーザー、引き受けたロールユーザー(assumed-role user)をARN形式で記述
  - “Principal” : { “AWS” : “arn:aws:iam::*123456789012*:/root”}
  - “Principal” : { “AWS” : “arn:aws:iam::*123456789012*:/user/*Alice*”}
  - “Principal” : { “AWS” : “arn:aws:iam::*112233445566*:/role/*s3ReadOnlyRole*”}
  - “Principal”: { “AWS” : "arn:aws:sts::*222255558888*:assumed-role/*role-name*/*role-session-name*" }

  注：IAMグループの指定は不可、大文字小文字は区別される、ユーザーを指定する際に“すべてのユーザー” の意味でワイルドカード (*) を使用することはできない
- IAMロールの信頼ポリシーのPrinicpal要素に指定したIAMユーザーとIAMロールを削除すると信頼関係は壊れる。
  - 同じ名前でIAMエンティティを作成してもプリンシパルIDが異なるため、同じ名前で再作成した場合はロールの再編集が必要

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements_principal.html

Action要素

- 許可/拒否される特定のアクションを指定する

- Statement要素にはAction/NotAction要素が必須
- AWSサービスで行うことができるタスクを記述する独自のアクションセットを記述
  - 有効なアクション名はドキュメントを参照、またはポリシーエディターから選択
  - 形式:“Action” : “<各サービスの名前空間>:<アクション名>”
  - “Action” : "ec2:StartInstances”
  - “Action” : [“sqs:SendMessage” , “sqs:ReceiveMessage”]
  - “Action” : "iam:*AccessKey”
  - “Action” : "IAM:listaccesskeys”

  注：複数のアクションを指定可能、ワイルドカード(*)を使用可能、値は大文字小文字の区別なし

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements_action.html

Resource要素

- ステートメントで取り扱う一連のオブジェクトを指定する

- Statement要素にはResource/NotResource要素が必須
- AWSサービスが持つ一連のリソースセットをARN形式で記述
  - 有効なリソースはドキュメントを参照、またはポリシーエディターから選択
  - "Resource" : "arn:aws:sqs:us-east-2:*123456789012*:*queue1*"
  - "Resource" : "arn:aws:iam::*123456789012*:user/accounting/*"
  - "Resource" : ["arn:aws:dynamodb:us-east-2:*123456789012*:table/*books_table*", "arn:aws:dynamodb:us-east-2: *123456789012* :table/*magazines_table*"]
  - "Resource" : "arn:aws:dynamodb:us-east-2:*123456789012*:table/${aws:username}"

  注：複数のリソースを指定可能、ワイルドカード(*)を使用可能、JSONポリシー変数を指定可能

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements_resource.html

Condition要素 (1/2)

- ポリシーが有効になるタイミングの条件を指定する

- Condition要素の記述はオプション
- 条件キー:条件値に対する評価方法として条件演算子を作用させる演算式を記述
  - 形式：”Condition” : { <条件演算子> : { <条件キー> : <条件値> } }
  - 条件演算子
    - 条件比較のタイプ (文字列条件、数値条件、IPアドレス条件等)を指定する
    - 条件キーごとに使用できる条件演算子の種類が決まっている
  - 条件キー
    - AWSグローバル条件コンテキストキー (”aws:”で始まる)
      - 全てのサービスで使用可能なキー、一部のサービスでのみ使用可能なキーがある
    - AWSサービス固有のキー (そのサービス固有の名前 (“s3:”等) で始まる)
    - IAMの条件コンテキストキー

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements_condition.html

Condition要素 (1/2)

- “Condition” : { “StringEquals” : { “aws:username” : “*johndoe*” } }
- “Condition” : { “StringEqualsIgnoreCase” : { “aws:username” : “johndoe” } }
- “Condition“ : { ”IpAddress” : { ”aws:SourceIP” : [“*192.0.2.0/24*” , “*203.0.113.0/24*”] } }
- “Condition“ : { ”StringEquals” : { ”ec2:ResourceTag/*tagkey*” : “*tagvalue*” } }
- "Condition" : { "StringEquals" : { "s3:prefix": "projects" } }
- “Condition” : { “StringEquals” : { “iam:PassedToService” : “cloudwatch.amazonaws.com” } }
- “Condition” : { "ForAllValues:StringEquals": { "dynamodb:Attributes": ["*ID*", "*Message*", "*Tags*"] }

注：条件キーは大文字小文字は区別しない、条件値の大文字小文字の区別は使用する条件演算子によって異なる

- 参考情報

  条件演算子

  https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html

  グローバル条件キー

  https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_condition-keys.html

  IAMの条件コンテキストキー

  https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_iam-condition-keys.html

要素のAND条件とOR条件

Condition下のブロックはAND条件、演算子に対する値はOR条件

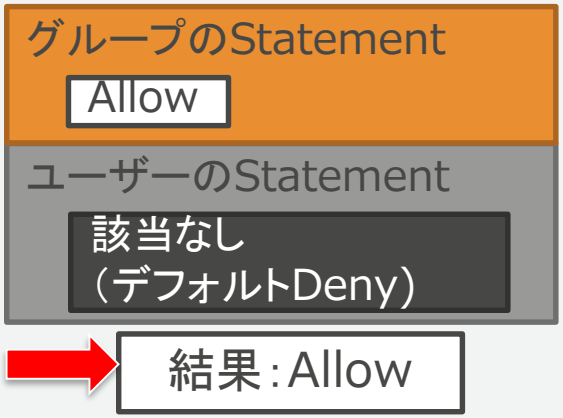
この例の場合、”2019/1/29の12:00から15:00の間に、ソースIPアドレス192.168.176.0/24もしくは192.168.143.0/24のネットワークからアクセスしたリクエスト”を意味する

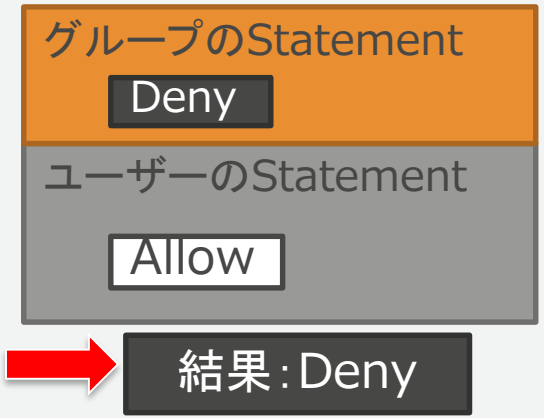
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_elements_condition.html

アクセス可否の決定ロジック

- **暗黙的なDeny (デフォルト) < 明示的なAllow < 明示的なDeny**

- すべてのアクセスはデフォルトで拒否 (**暗黙的なDeny**)
- アクセス権限に“Allow”の条件があった場合、アクセス許可
- ただしアクセス権限に１つでも“Deny”の条件があった場合、アクセス拒否(明示的なDeny)





https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies_evaluation-logic.html

IAMポリシーの作成を支援するツール群

- ビジュアルエディター機能
  - 最初から新しいポリシー構築可能
- AWS Policy Generator：http://awspolicygen.s3.amazonaws.com/policygen.html
  - AWSのサービスについて、必要情報を入力するとポリシー文書を自動作成してくれるツール
- ポリシー言語の文法チェック機能
  - ポリシー保管時にポリシー言語の文法チェック、自動フォーマットを実施
  - 「Validate Policy」により明示的な確認が可能
- IAM Policy Validator
  - 自動的に既存の IAMポリシーを調べ、IAMポリシーの文法に準拠しているか確認
  - ポリシーに対する推奨の変更を提示
  - Policy Validator を使用できるのは、準拠していないポリシーがある場合のみ
- IAM Policy Simulator：https://policysim.aws.amazon.com/home/index.jsp
  - プロダクションへの実装前にポリシーをテスト可能
  - パーミッションのトラブルシューティング
  - Condition、ポリシー変数、リソースベースのポリシーを入れたテスト

✓ 追加セキュリティに対するポリシー条件を使用する

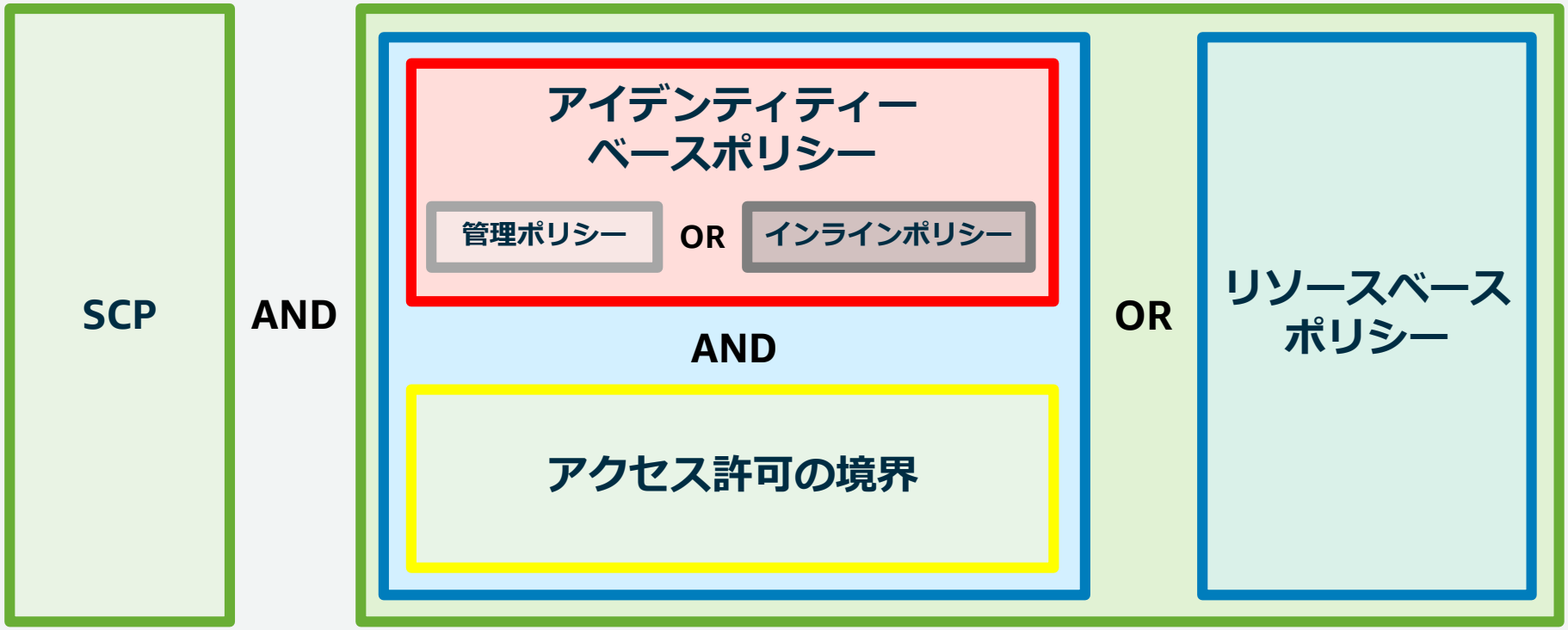
Use Policy Conditions for Extra Security

- より安全なポリシーの適用のために、Condition要素によってポリシーが有効になる条件をさらに絞り込む
  - リクエストを許容するソースIPアドレスの範囲
  - 日付または時間の範囲
  - MFAデバイスでの認証の要求
  - SSL使用の要求

アクセス権限の管理

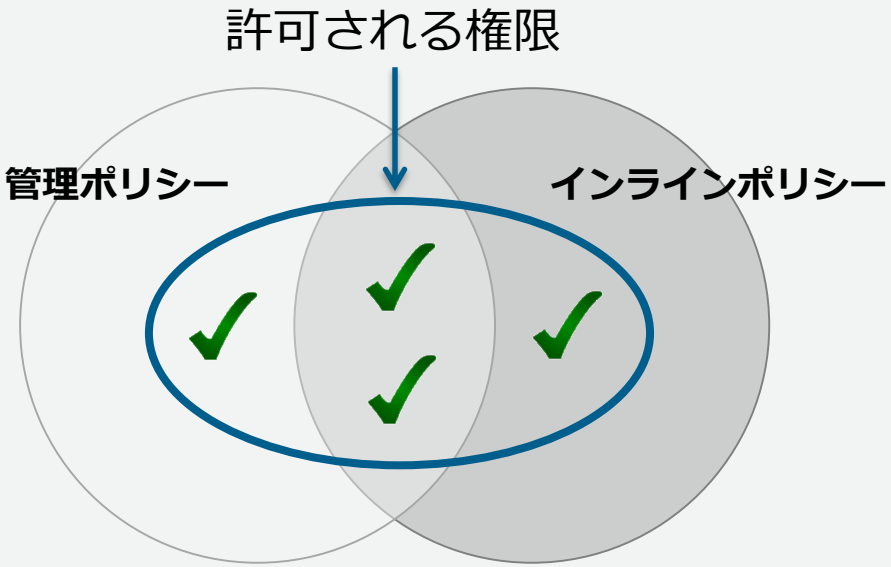
- ✓ AWS管理ポリシーを使用したアクセス許可の使用開始
- ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する
- ✓ 追加セキュリティに対するポリシー条件を使用する
- ✓最小権限を付与する
- ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する

アクセス権の決定ロジック (同一アカウントの場合)



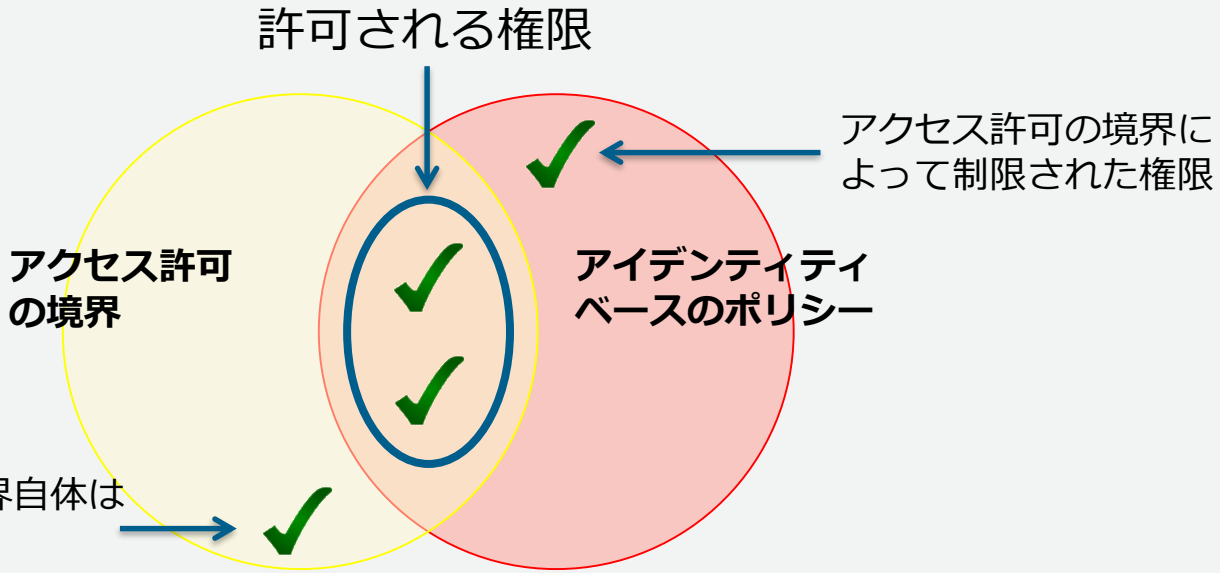
アイデンティティベースのポリシー

- 管理ポリシーとインラインポリシーのそれぞれで許可されているものが有効な権限となる (OR条件)



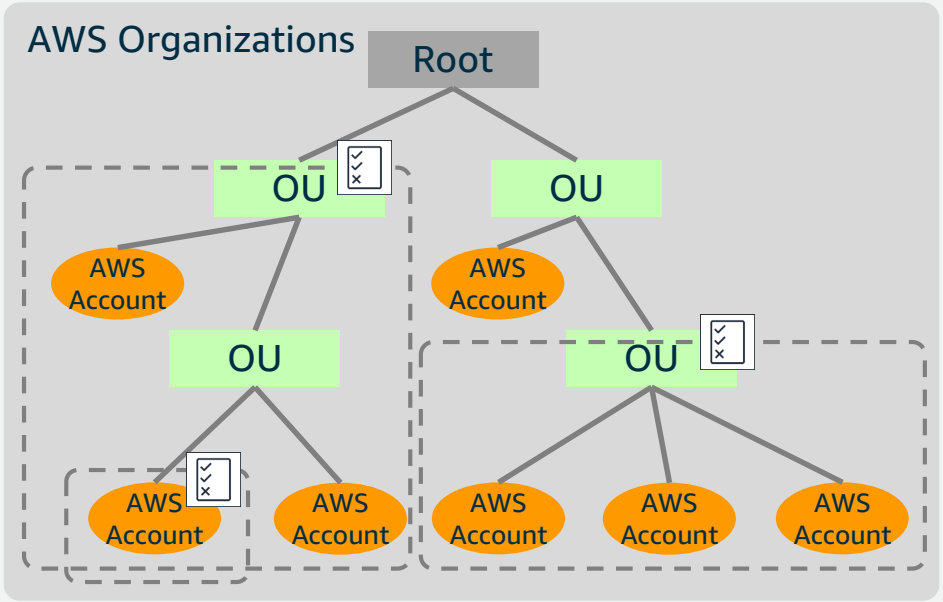
アクセス許可の境界 (Permission Boundary)

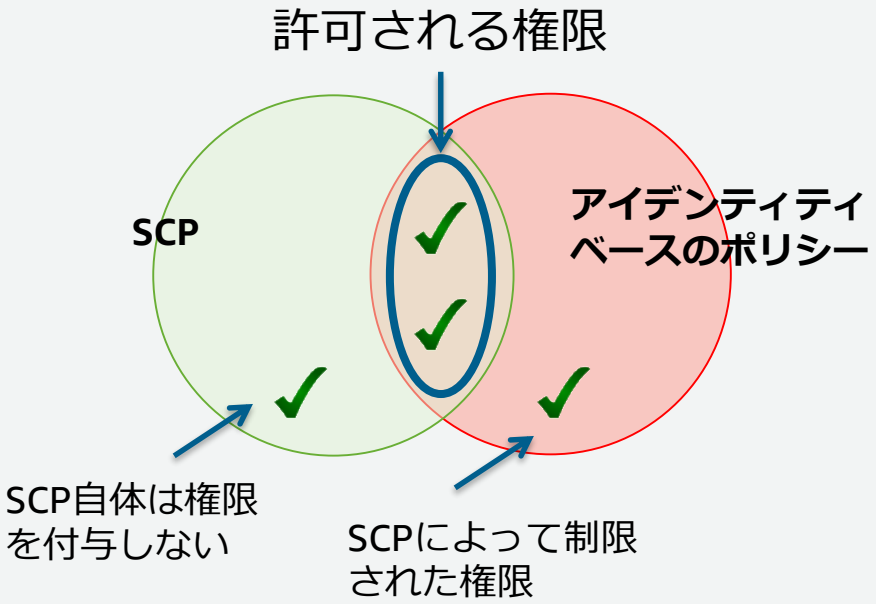
- アクセス許可の境界とアイデンティティベースのポリシーの両方で許可されているものが有効な権限となる (AND条件)



AWS Organizations サービスコントロールポリシー (SCP)

- SCPとアイデンティティベースのポリシーの両方で許可されているものが有効な権限となる (AND条件)

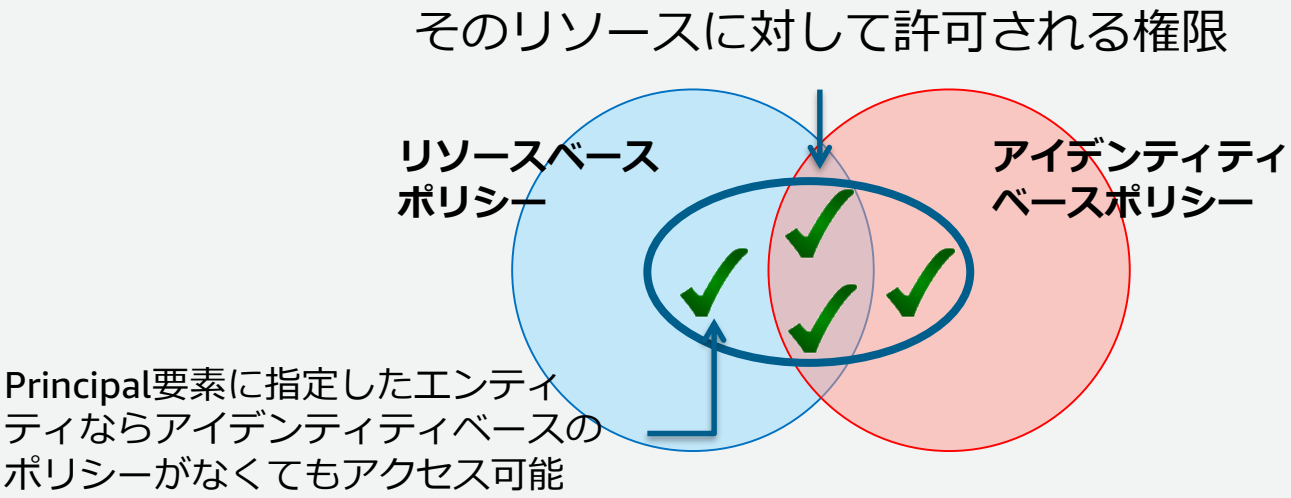




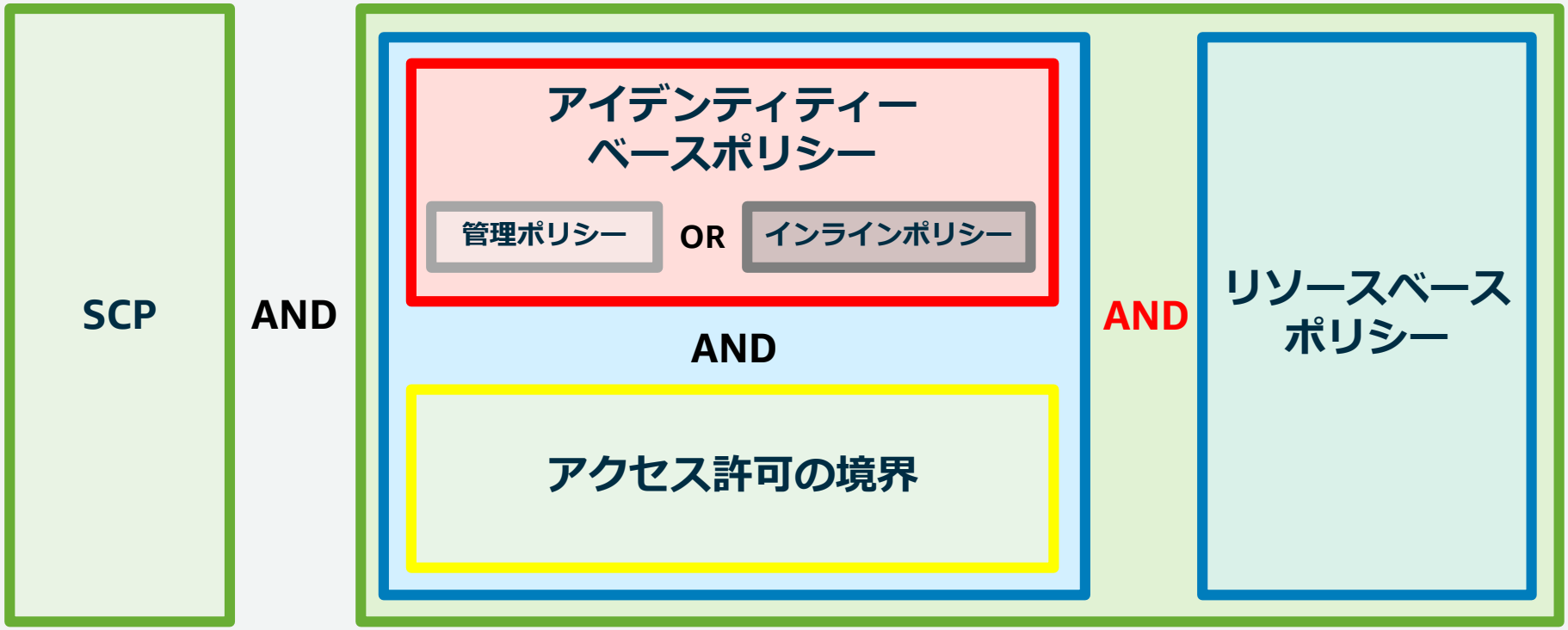
https://d1.awsstatic.com/webinars/jp/pdf/services/20180214_AWS-Blackbelt-Organizations.pdf

リソースベースポリシー (同一アカウントの場合)

- 同一アカウントの場合、リソースベースポリシーとアイデンティティベースポリシーのそれぞれで許可されているものが有効となる (OR条件)

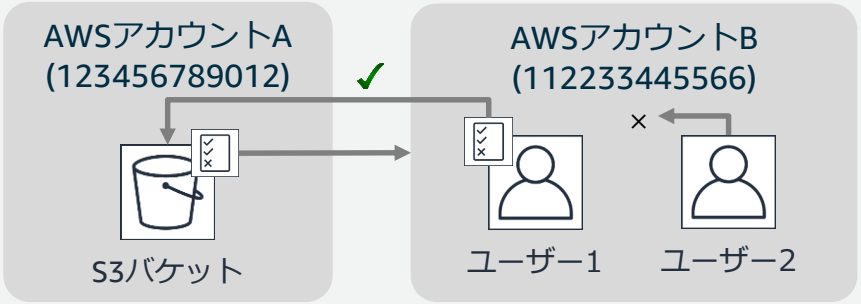


アクセス権の決定ロジック (クロスアカウントの場合)

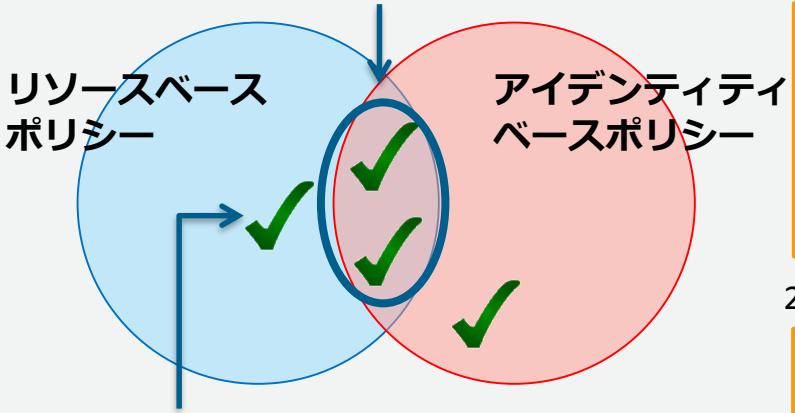


リソースベースポリシー (クロスアカウントの場合)

- クロスアカウントの場合、リソースベースポリシーとアイデンティティベースポリシーの両方で許可されているものが有効となる (AND条件)



そのリソースに対して許可される権限



Principal要素に信頼するエンティティの指定が必要

1.AWSアカウントAのバケットポリシーに以下の権限を設定

```
{
  "Statement" : {
    "Effect":"Allow",
    "Principal" :{“AWS”:“arn:aws:iam::112233445566:root"},
    "Action":"s3:*",
    "Resource":"arn:aws:s3:::mybucket/*"
  }
}
```

Principalは、実行をしているユーザーに対する条件設定

2.AWSアカウントBのエンティティにアクセス権限を付与

```
{
  "Statement" : {
    "Effect":"Allow",
    "Action":”s3:GetObject”,
    "Resource":"arn:aws:s3:::mybucket/*"
  }
}
```

✓ 最小権限を付与する

Grant Least Privilege

- IAMポリシーを作成する場合、タスクの実行に必要なアクセス許可のみ付与する

- 最小限のアクセス権限から開始し、必要に応じてアクセス権限を追加する
  - あとでアクセス権限を強化するより安全なアプローチ
- 役立つ情報
  - アクセスアドバイザーのサービスの最終アクセス時間データ
  - Amazon CloudTrailのイベントログ
  - IAMポリシーのリファレンス

    https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_policies.html
  - Blackbeltオンラインセミナー AWSサービスの権限管理

    https://d1.awsstatic.com/webinars/jp/pdf/solution-casestudy/20160621_AWS-BlackBelt-Authority-public.pdf

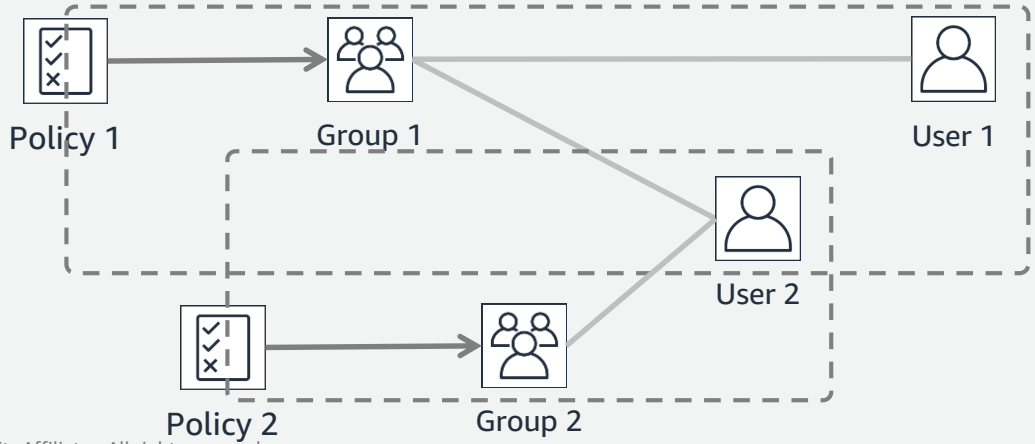
アクセス権限の管理

- ✓ AWS管理ポリシーを使用したアクセス許可の使用開始
- ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する
- ✓ 追加セキュリティに対するポリシー条件を使用する
- ✓ 最小権限を付与する
- ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する

IAMグループ

- IAMユーザーの集合

- IAMグループやIAMロールをIAMグループに所属させることは不可
- IAMユーザーは複数のIAMグループに所属することができる (最大10)
- IAMグループに関連付けられたIAMポリシーは所属するIAMユーザーに継承される



✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する Use Groups to Assign Permissions to IAM Users

- ポリシーの関連付けを簡単にするためにはIAMグループを利用した方が便利

- 組織またはジョブ機能に関連したIAMグループを作成し、IAMグループに対してアクセスIAMポリシーを関連付ける
- 会社内で組織異動がある場合は、そのIAMユーザーが所属するIAMグループを変更すればよい

ここまでのまとめ：アクセス権限の管理のベストプラクティス

- ✓ AWS管理ポリシーを使用したアクセス許可の使用開始
- ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する
- ✓ 追加セキュリティに対するポリシー条件を使用する
- ✓ 最小権限を付与する
- ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する

まとめ

AWS IAMのベストプラクティス

IDと認証情報の管理	✓ AWSアカウントのルートユーザーアクセスキーをロックする ✓ 個々のIAMユーザーを作成 ✓ ユーザーの強力なパスワードポリシーを設定 ✓ アクセスキーを共有しない ✓ 特権ユーザーに対してMFAを有効化する
アクセス権限の管理	✓ AWS管理ポリシーを使用したアクセス許可の使用開始 ✓ インラインポリシーではなくカスタマー管理ポリシーを使用する ✓ 追加セキュリティに対するポリシー条件を使用する ✓ 最小権限を付与する ✓ IAMユーザーへのアクセス許可を割り当てるためにグループを使用する
権限の委任	✓ Amazon EC2インスタンスで実行するアプリケーションに対し、ロールを使用する ✓ ロールを使用したアクセス許可の委任
IDと権限のライフサイクル管理	✓ AWSアカウントのアクティビティの監視 ✓ アクセスレベルを使用して、IAM権限を確認する ✓ 不要な認証情報を削除する ✓ 認証情報を定期的にローテーションする

Part 2

https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/best-practices.html

まとめ

- AWS IAMはAWSサービスを利用するための認証と認可を提供する
- 最小限の認証情報の生成、強力なパスワードポリシー、特権ユーザーでのMFA有効化がアイデンティティの保護に有効
- ポリシードキュメントとポリシーの論理評価ロジックに対する理解を深めて、最小権限を追求

参考情報へのリンク

- AWS IAM 公式サイト
https://aws.amazon.com/jp/iam/

- AWS IAMドキュメント
https://docs.aws.amazon.com/iam/index.html

- AWS Security Blog
http://blogs.aws.amazon.com/security/

- IAMの制限 (IAMドキュメント)
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_iam-limits.html

- AWSアカウントの認証管理 (AWS Summit Tokyo 2018)
https://d1.awsstatic.com/events/jp/2018/summit/tokyo/aws/40.pdf

- AWSご利用開始時に最低限おさえておきたい10のこと (Blackbelt Online Semminer)
https://d1.awsstatic.com/webinars/jp/pdf/services/20180403_AWS-BlackBelt_aws10.pdf

Q&A

お答えできなかったご質問については
AWS Japan Blog 「https://aws.amazon.com/jp/blogs/news/」にて
資料公開と併せて、後日掲載します。

ご視聴ありがとうございました

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive