アイデンティティラウンドロビン ワークショップ

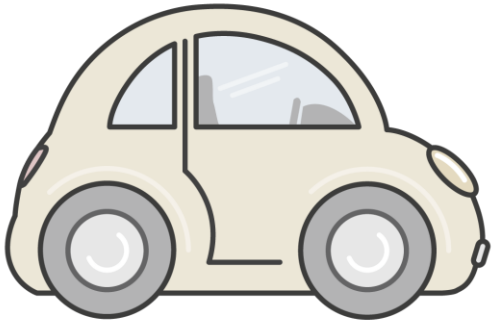
パーミッションバウンダリー

アジェンダ

- パーミッションバウンダリーの概要と基本事項
- デモ
- 許可カテゴリー
- パーミッションバウンダリーの仕組み
- リソースの制限

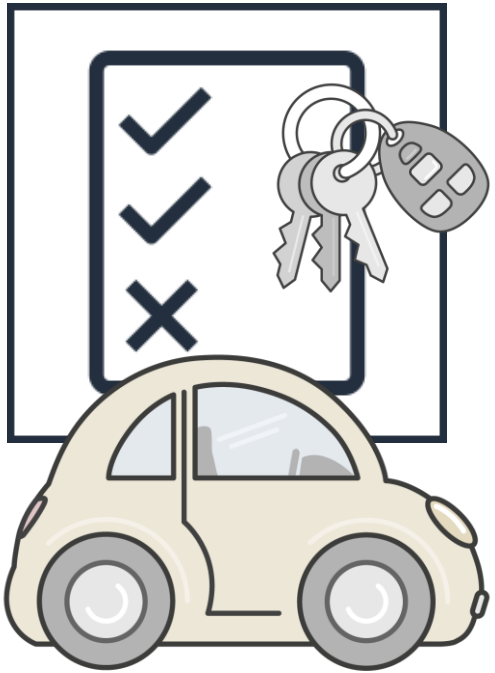
アナロジー – ティーンエイジャーに鍵を渡す

- 車のキーがあればいろいろなことができます。車を飛ばしてどこへでも行けます。飲酒運転だってできます。
- スピード違反をしない、20 マイルより遠いところに行かないなどの規則を定めることはできますが、信頼だけが頼りです。
- もう 1 つのオプションは、発見的統制とコンプライアンス (走行距離計の確認、スピード違反チケットをもらっていないか、事故を起こしていないかの確認) のみです。



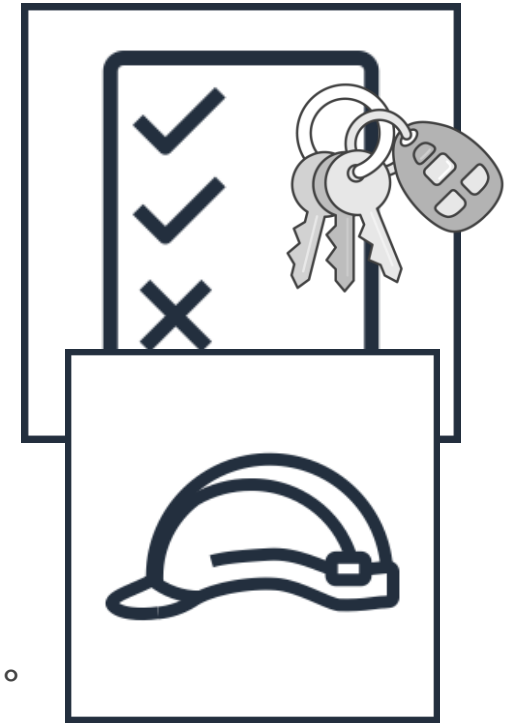
アナロジー – ティーンエイジャーに鍵を渡す

- 制限を設定して運転を許可できるような特別な鍵とプログラミングが可能な車種もあります。
- 車でできること (車を飛ばす、ラジオを大音響で鳴らす、タイヤをスピンさせる) は、ティーンエイジャーの欲望とあなたの設定の交点で決まります。



アナロジー – 開発者に鍵を渡す

- 同じように、鍵 (ユーザーやロールを作成する能力) およびそれに付随するすべての権限を開発者に与えることができます。
- 開発者は、完全な管理者権限が設定されたアイデンティティベースのポリシーをロールにアタッチできますが、アクセス許可の境界 (車の制限設定に相当) もアタッチする必要があります。
- ロールで有効な許可はこれら 2 つの交点になります。



パーミッションバウンダリーとは

ユーザーやロールを作成する許可を委任する仕組みで、権限のエスカレーションを防止し、不必要に広範囲の許可を与えないようにします。
ユーザーやロールが取得できる最大許可を制御しますが、許可自体は付与しません。

次のようなアクションを安全に付与する方法です。
"iam:CreateRole"
"iam:PassRole"

パーミッションバウンダリーの使用前と使用後

使用前

- 特定の IAM ポリシーアクション (PutUserPolicy、AttachRolePolicy など) には基本的には完全に管理者と同様の許可が付与されていました。
- セルフサービス許可管理の実行はいずれも煩雑でした。

使用後

- 管理者は完全に管理者と同様の許可を付与できますが、"アクセス許可の境界" を指定します。
- 開発者はアプリケーションのプリンシパルを作成してポリシーをアタッチできますが、自分の境界内のみに限定されます。

aws

ユースケース

- Lambda 関数のロールを作成する必要がある開発者
- EC2 インスタンスのロールを作成する必要があるアプリケーション所有者
- 特定のユースケースのユーザーを作成する必要がある管理者
- その他

パーミッションバウンダリーの基本事項

パーミッションバウンダリーのワークフロー

管理者

委任対象の管理者の作成

要件: 委任対象の管理者が作成するユーザーとロールにはパーミッションバウンダリーの設定が必要

委任対象の管理者

"制限された" ユーザーとロールの作成

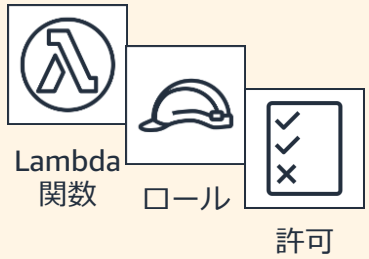
能力:パーミッションバウンダリーがアタッチされたユーザーとロールを作成可能

"制限された" IAM のユーザーとロール

パーミッションバウンダリーで制限されたユーザーとロール

結果: パーミッションバウンダリーにより、ユーザーとロールの許可を制限

制限対象リソース



制限対象リソースに対する許可

Lambda 関数などのリソースにアタッチされたロールの許可をパーミッションバウンダリーによって制限

IAM 条件コンテキストキーは...

```
"Condition": {"StringEquals":
    {"iam:PermissionsBoundary":
    "arn:aws:iam::ACCOUNT_ID:policy/permissionboundary"
    }
}
```

主要な作成アクション (ユーザーとロール) に適用されます

```
"Effect": "Allow",
"Action": ["iam:CreateRole"],
"Resource": ["arn:aws:iam::ACCOUNT_ID:role/path/"],
"Condition": {"StringEquals":
    {"iam:PermissionsBoundary":
    "arn:aws:iam::ACCOUNT_ID:policy/permissionboundary"
    }
}
```

エンドユーザー体験

Lambda 関数のロールの作成

 # ステップ 1: ロールの作成とアクセス許可の境界のアタッチ

```
$ aws iam create-role –role-name roleforlambda
–assume-role-policy-document file://Role_Trust_Policy_Text.json
–permissions-boundary arn:aws:iam::<ACCOUNT_NUMBER>:policy/department_a/boundary_1
```

 # ステップ 2: アイデンティティベースのポリシーの作成

変更なし

 # ステップ 3: アイデンティティベースのポリシーのアタッチ

変更なし

デモ

- **ユーザー要件:**
  - S3 バケットを読み取る Lambda 関数
  - Lambda 関数にはバケットにアクセスする IAM ロールが必要
  - 正しい許可を設定したロールの作成が必要
- **企業の要件**:
  - ロールにアタッチされるポリシーでは、権限のエスカレーションや不必要な許可を禁止
  - ユーザーの妨げにならないこと

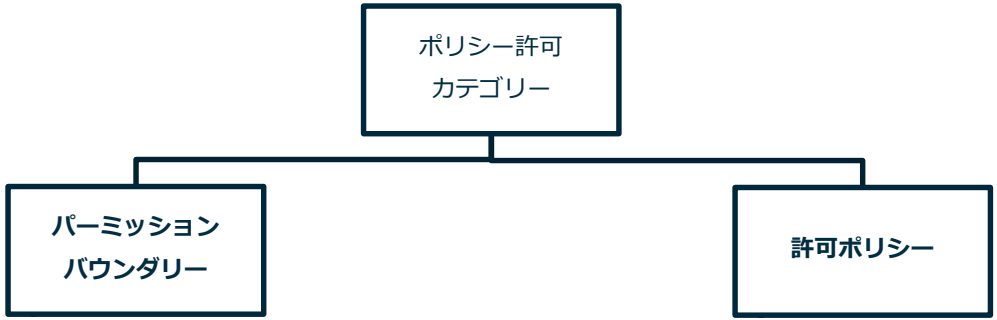
作成:
- **ユーザーのポリシー (+ 読み取り専用ポリシー)**
- **パーミッションバウンダリーのポリシー**
- **ユーザー**

作成:
- **ロールのポリシー**
- **ロール**
- **Lambda 関数**

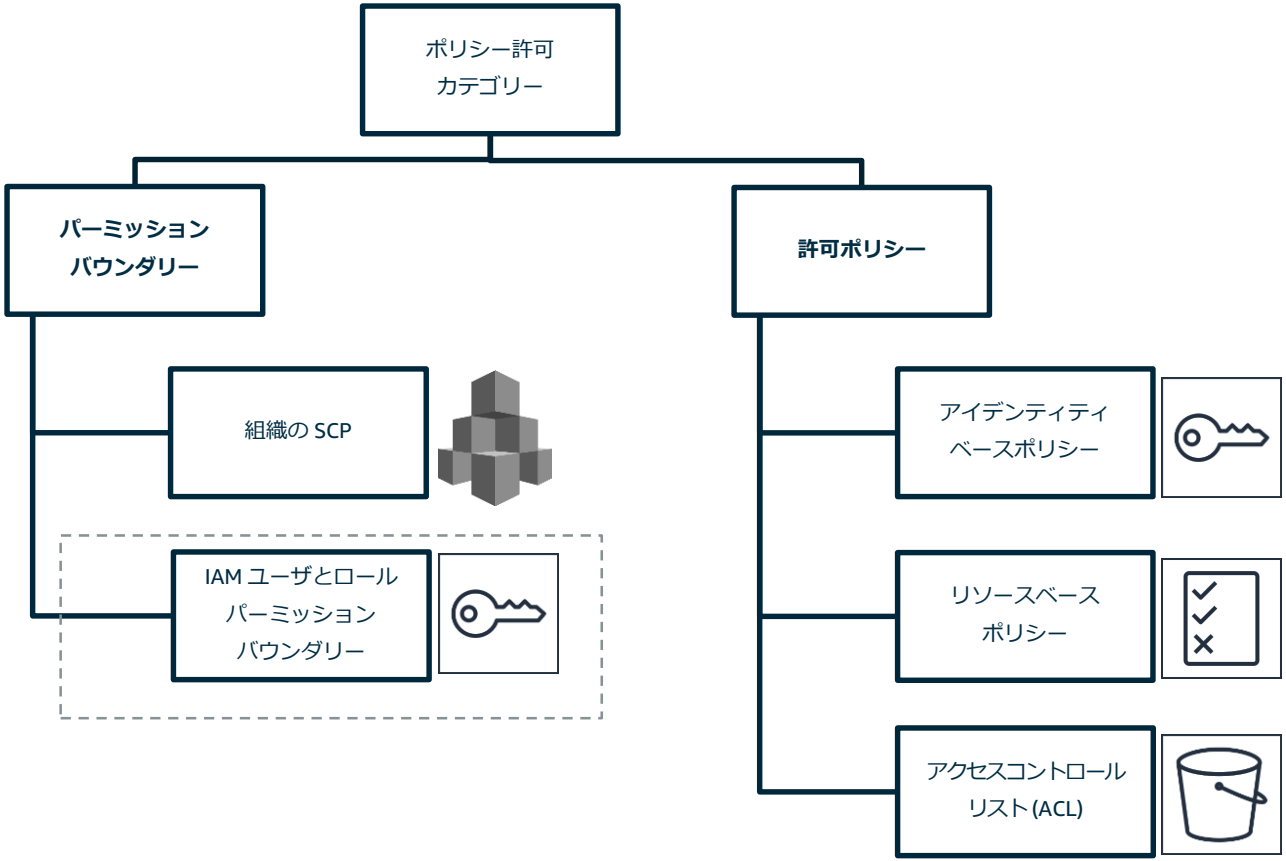
パーミッションバウンダリーで制限された Lambda 関数

ポリシーカテゴリー

ポリシー許可カテゴリー



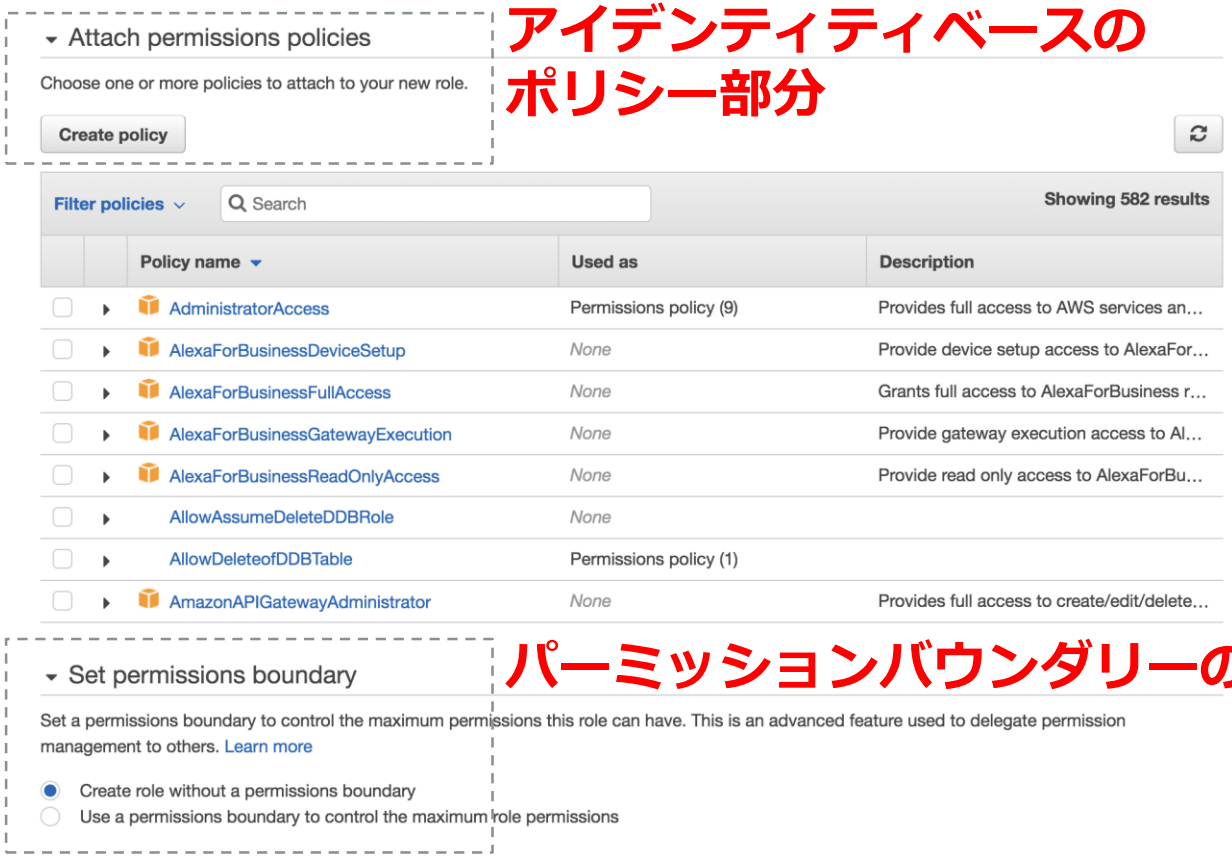
ポリシー許可カテゴリー



でも、単なる管理対象 IAM ポリシーでは?

IAM ポリシー

でも、単なる IAM ポリシーでは?

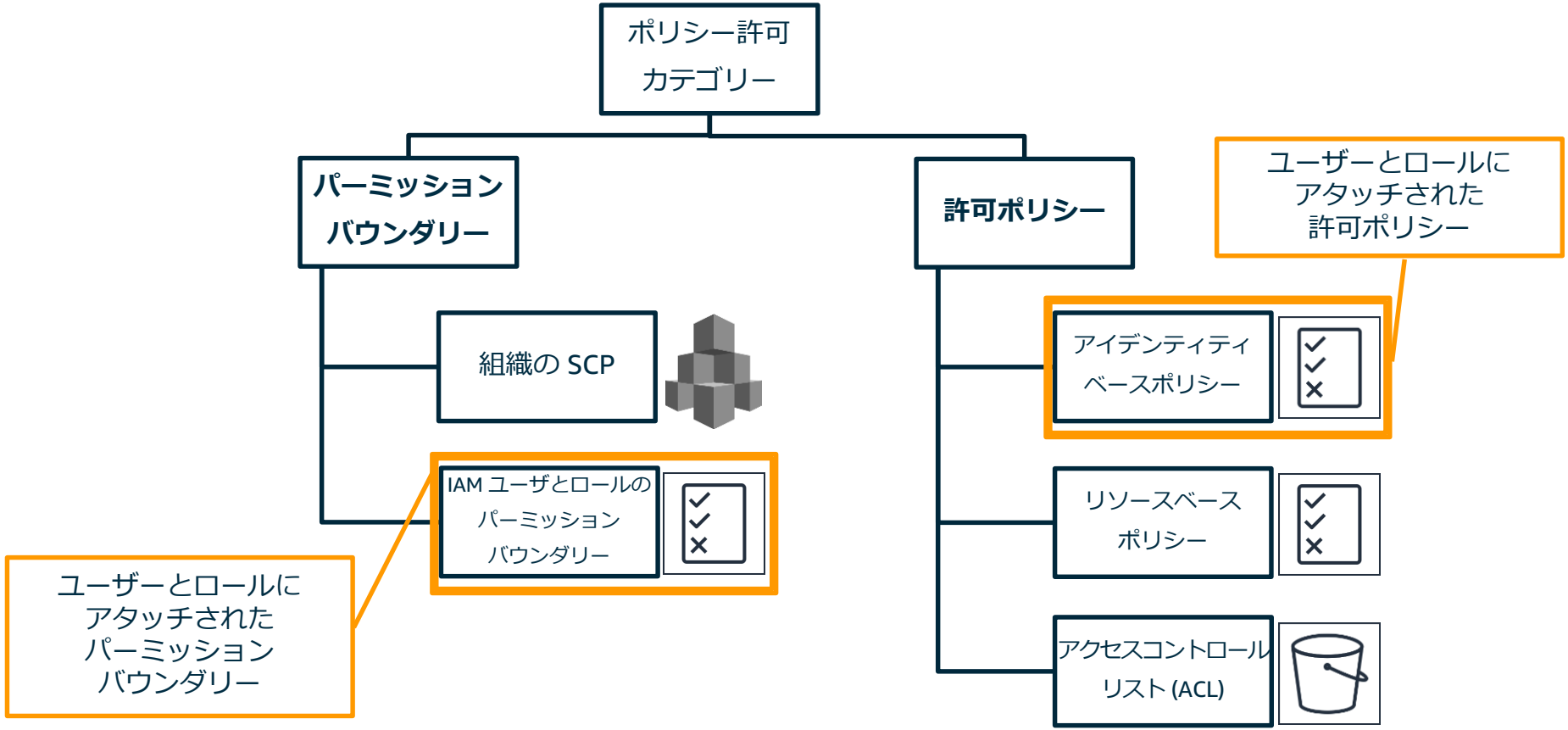


プレゼンテーションの質問 1

- パーミッションバウンダリーには、どのような条件コンテキストキーが使われていますか。
- パーミッションバウンダリーとアイデンティティベースのポリシーはどのように異なりますか。
- パーミッションバウンダリーのユースケースにはどのようなものがありますか。

パーミッションバウンダリーの仕組み

ポリシー許可カテゴリー



認証後の流れ

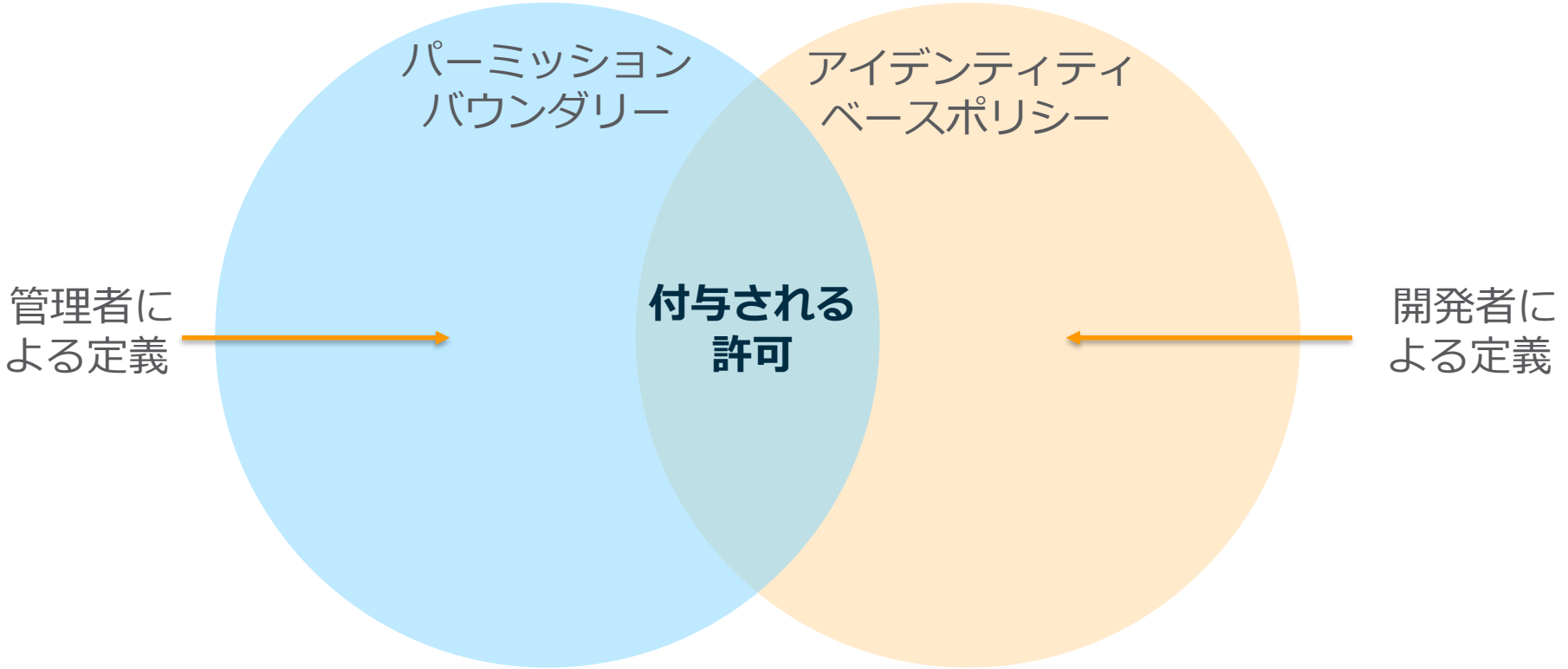
1. プリンシパルを**認証します。**

2. 要求にどの**ポリシー**を適用するかを決定します。

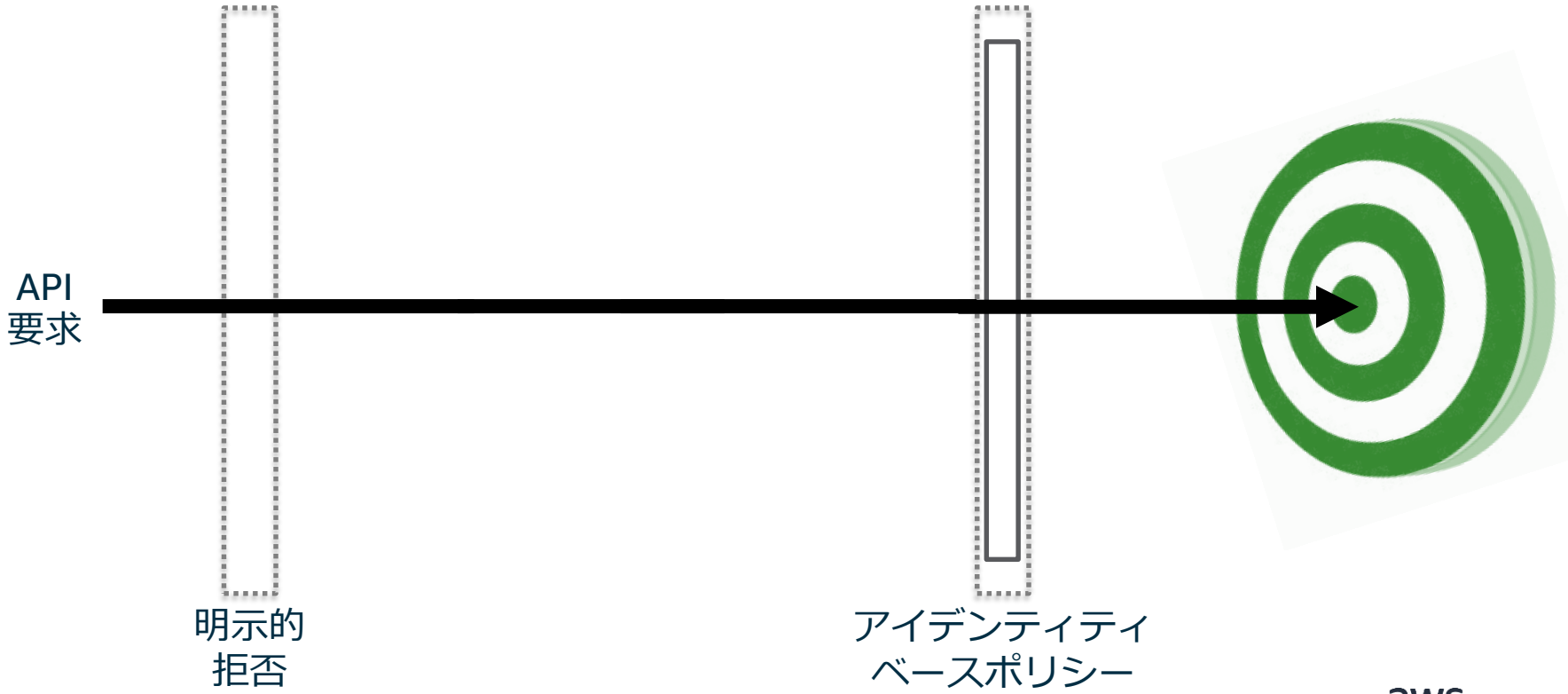
3. 該当する様々なポリシータイプを**評価**します。ポリシータイプを評価する順序はポリシータイプによって決まります。

4. 要求を**許可または拒否**します。

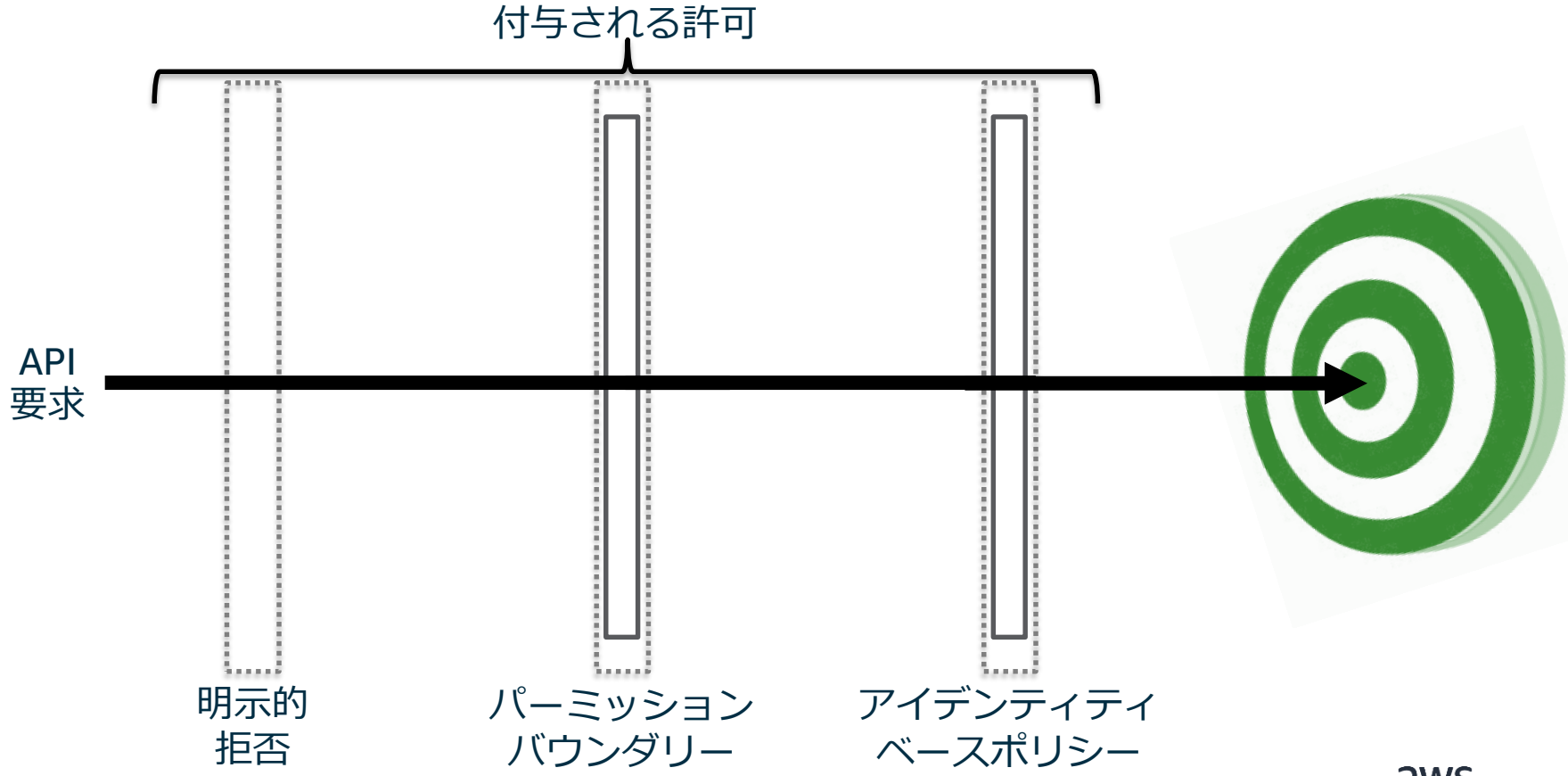
有効な許可 – ベン図



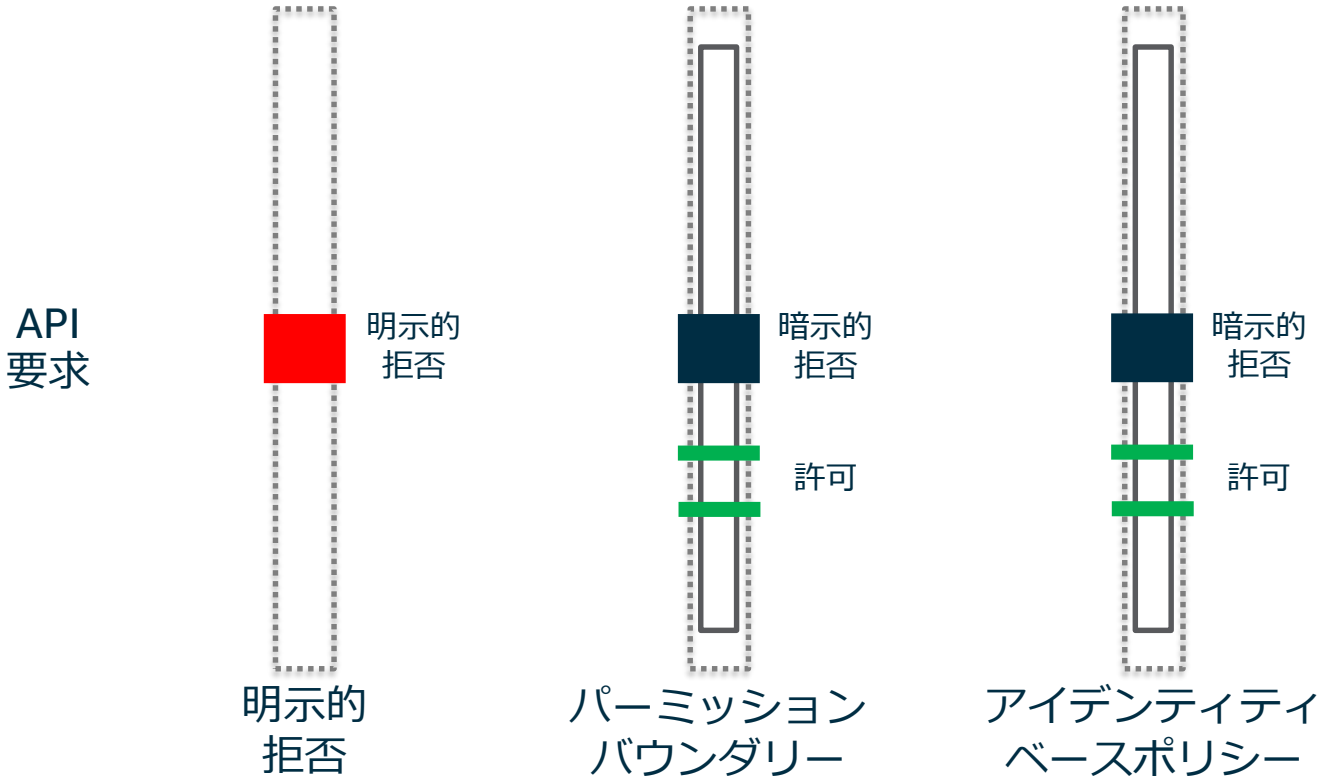
有効な許可 – 仕組み



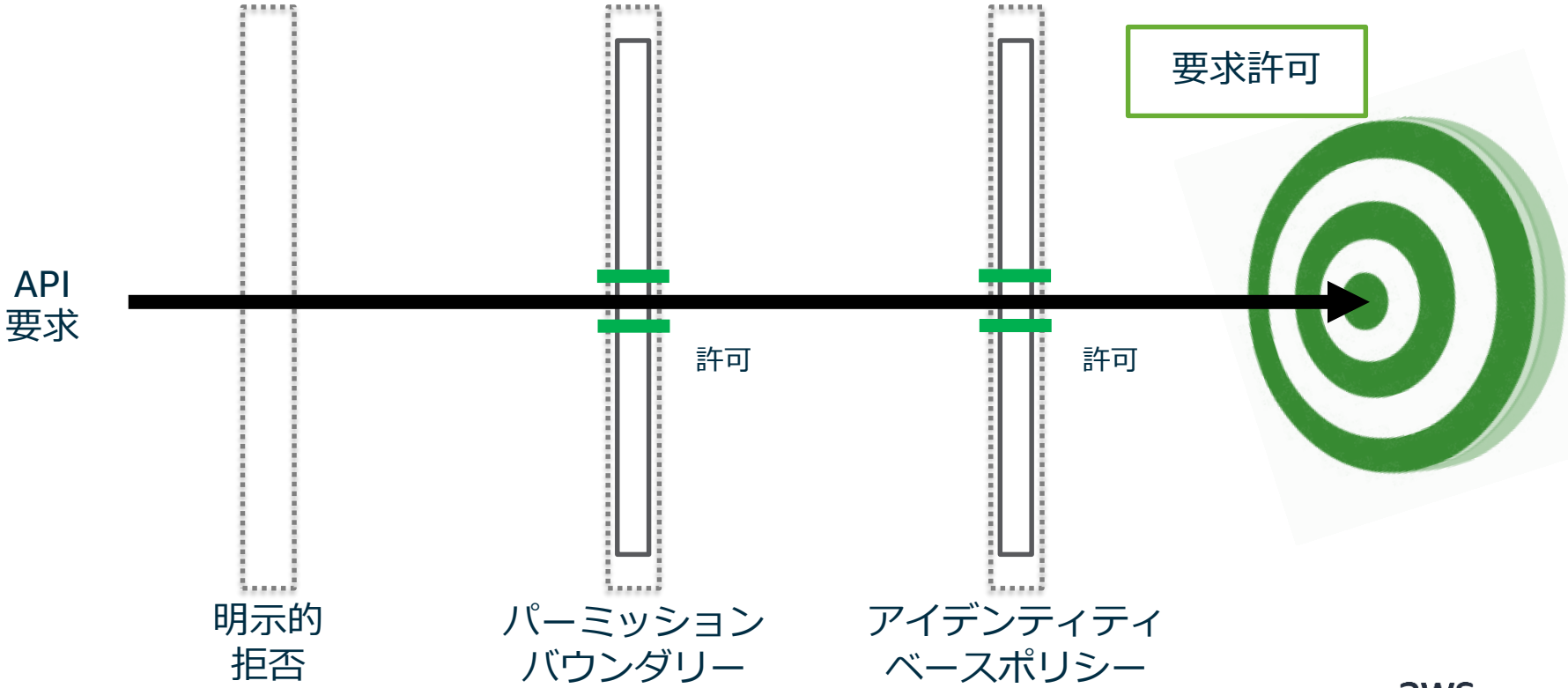
有効な許可 – 仕組み



有効な許可 – 仕組み



有効な許可 – 許可の例



有効な許可 – シナリオ 1

要求: s3:GetObject / バケット名: example1

パーミッションバウンダリー

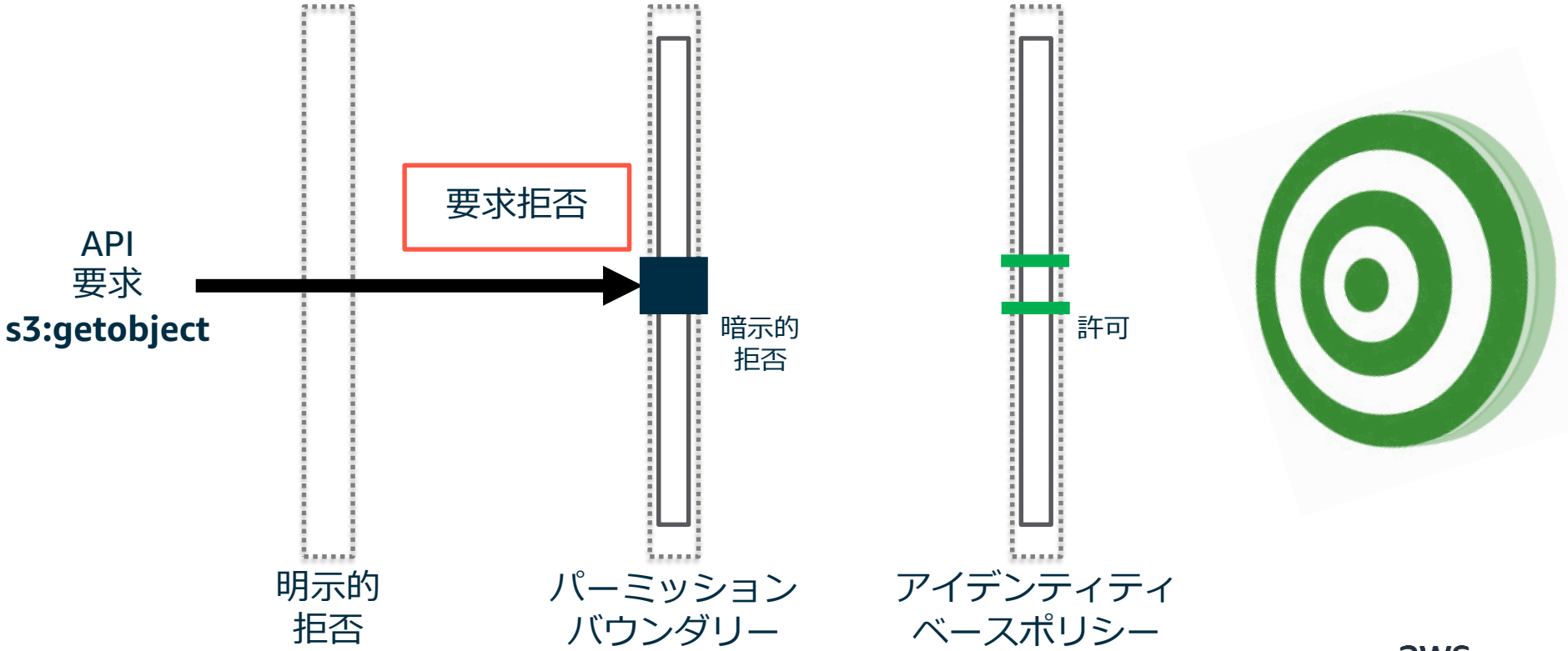
```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": [
         "logs:CreateLogGroup",
         "logs:CreateLogStream”,
         "logs:PutLogEvents”
       ],
     "Resource": "arn:aws:logs:*:*:*"
     }
}
```

アイデンティティベースポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": [
         "logs:CreateLogGroup",
         "logs:CreateLogStream",
         "logs:PutLogEvents",
         "s3:*"
        ],
      "Resource": "*"
      }
   ]
}
```


aws

有効な許可 – 結果



有効な許可 – シナリオ 2

要求: s3:GetObject / バケット名: example1

パーミッションバウンダリー

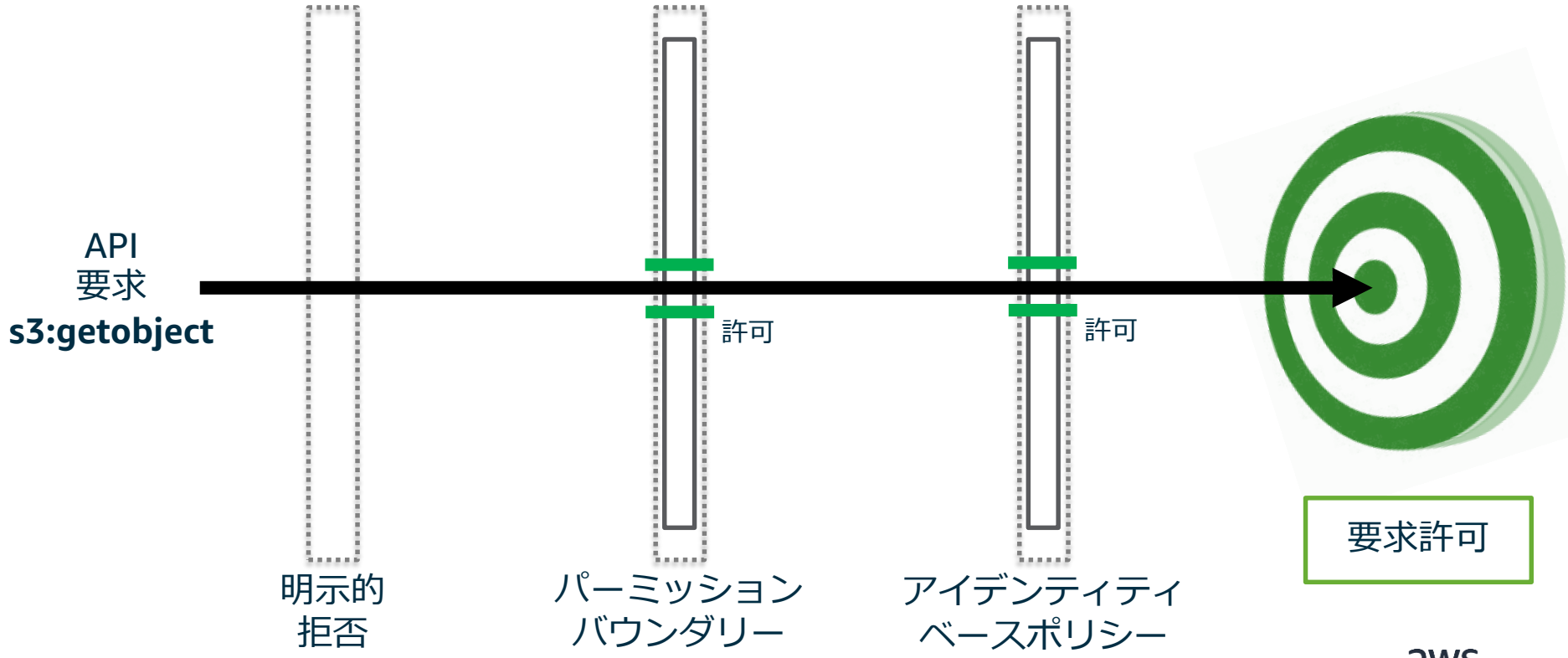
```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream”,
            "logs:PutLogEvents”
       ],
     "Resource": "arn:aws:logs:*:*:*"
     },
     {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource”:"arn:aws:s3:::example1/*"
     }
}
```

アイデンティティベースポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:PutLogEvents",
            "s3:*"
        ],
      "Resource": "*"
      }
   ]
}
```


aws

有効な許可 – 結果



有効な許可 – シナリオ 3

要求: s3:GetObject / バケット名: example1

パーミッションバウンダリー

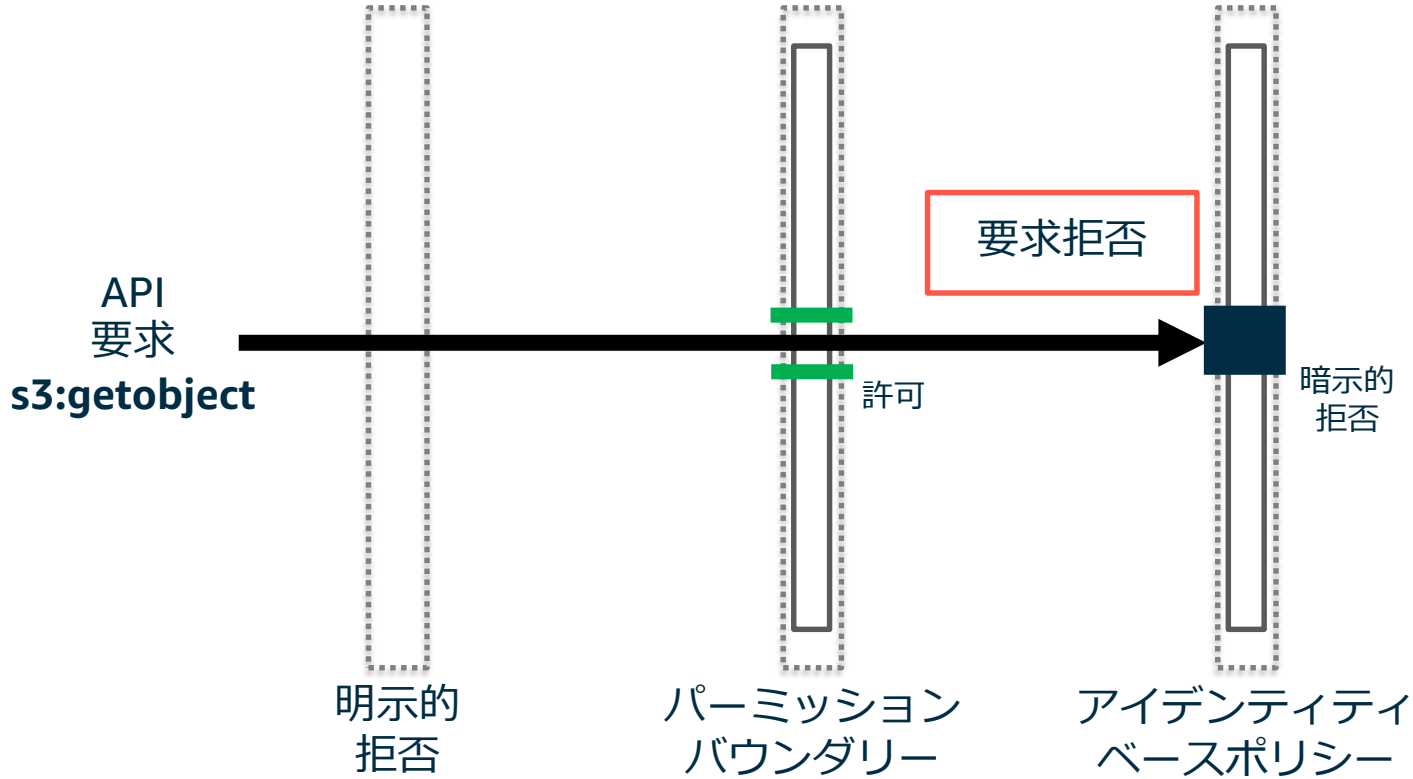
```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream”,
            "logs:PutLogEvents”
       ],
     "Resource": "arn:aws:logs:*:*:*"
     },
     {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource”:"arn:aws:s3:::example1/*"
     }
}
```

アイデンティティベースポリシー

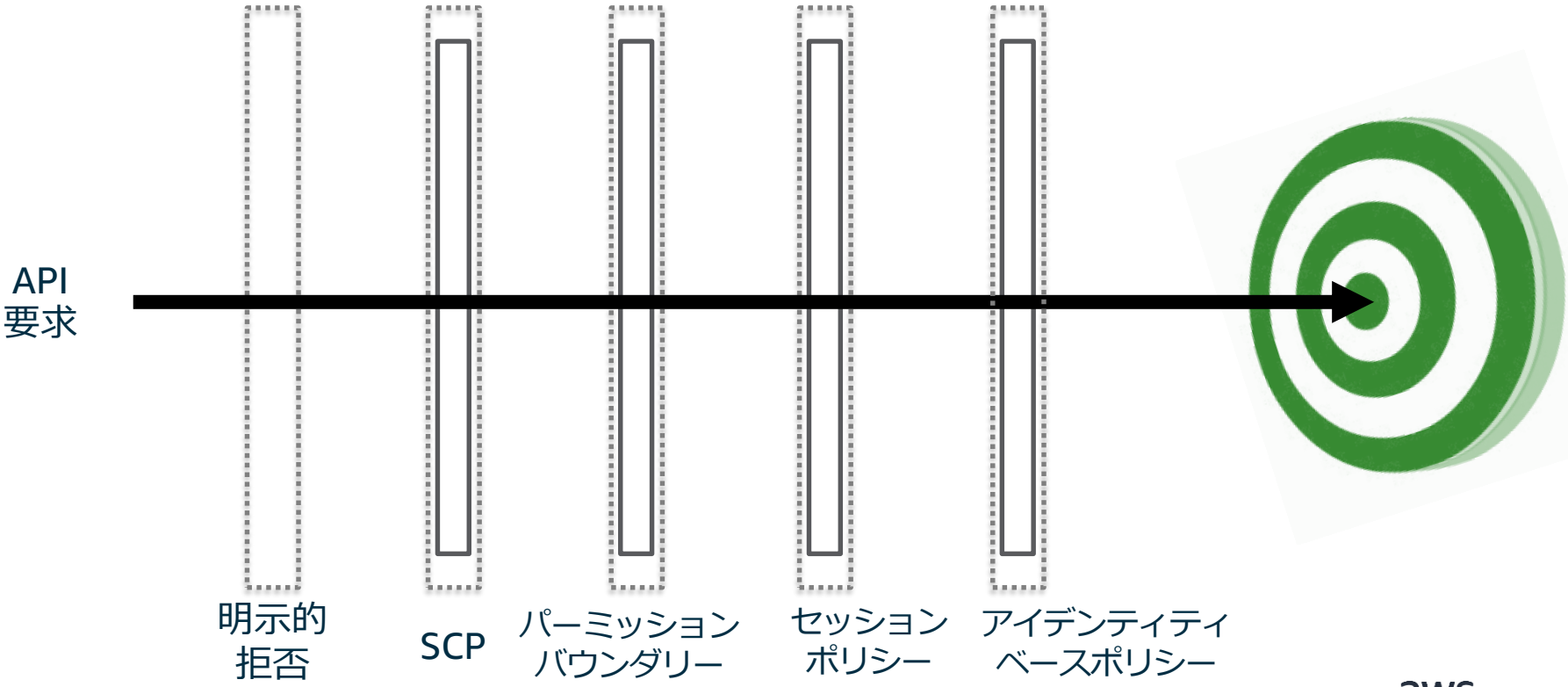
```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:PutLogEvents",
       ],
      "Resource": "*"
      }
   ]
}
```


aws

有効な許可 – 結果



有効な許可 – 仕組みの拡張



リソースの制限

目標: 他のリソースに影響を与えずに委任対象の管理者がリソースを変更できる余地を作り出すこと。

パスの使用が好ましいですが CLI が必要です。名前 (department1* など) も使用できます。

https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html#arns-paths

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_identifiers.html#identifiers-arns

リソースの制限 - 例

パスを使用したリソースの制限**:**

"Resource": "arn:aws:iam::123456789012:role**/department1/***"

ロールの例:

arn:aws:iam::123456789012:role/department1/role1

名前を使用したリソースの制限**:**

"Resource" : "arn:aws:iam::123456789012:policy**/development-users***"

ポリシーの例:

arn:aws:iam::123456789012:policy/development-users-policy1

aws

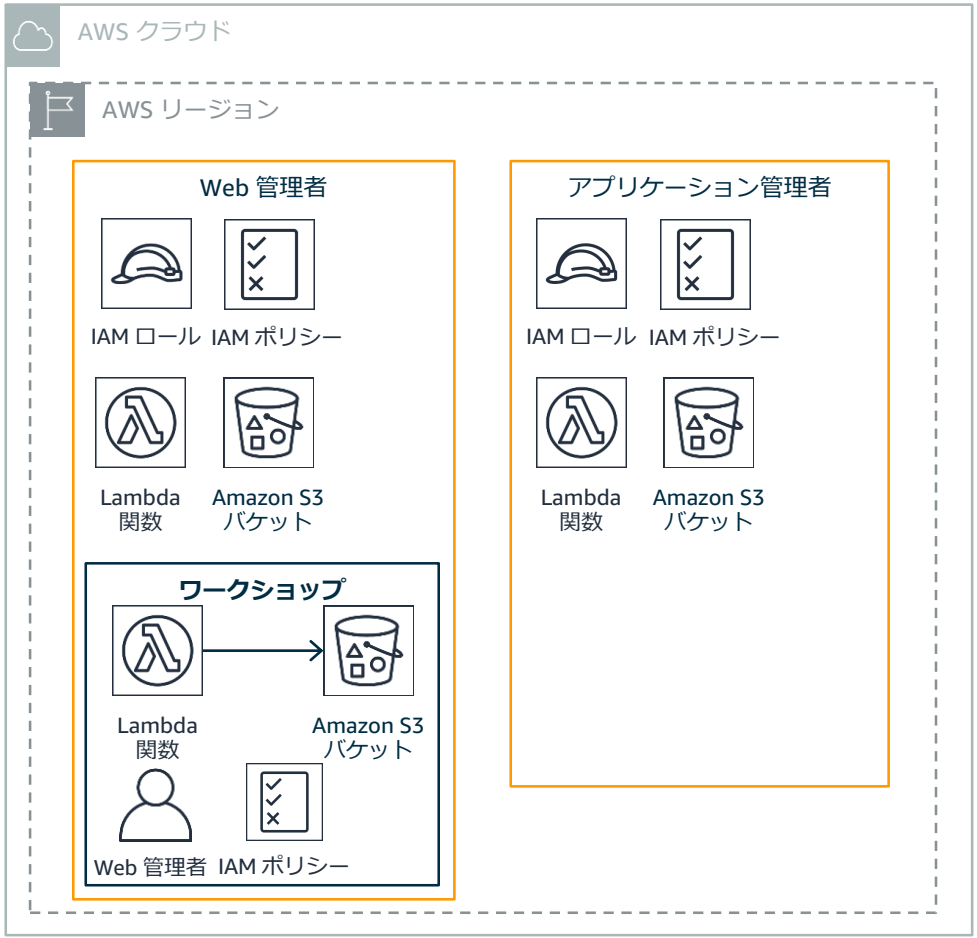
ワークショップ

このラウンドは**作成**と**検証**フェーズで構成されています。

作成 (60 分): まず各チームが作成フェーズに関連するアクティビティを実行します。

検証 (30 分): 各チームが Web 管理チームの一員であるかのように検証アクティビティを実行します。

ワークショップ


AWS クラウド
AWS リージョン
Web 管理者
IAM ロール
IAM ポリシー
Lambda 関数
Amazon S3 バケット
ワークショップ
Lambda 関数
Amazon S3 バケット
Web 管理者
IAM ポリシー
アプリケーション管理者
IAM ロール
IAM ポリシー
Lambda 関数
Amazon S3 バケット

パーミッションバウンダリー
作成フェーズ (60 分)

使用: 米国東部 (オハイオ)
us-east-2

ワークショップ

https://awssecworkshops.com

https://awssecworkshops.com/workshops/identity-round-robin/permission-boundaries/

[Overview] をクリックして CloudFormation テンプレートを実行し、[Build] をクリックしてください

aws

パーミッションバウンダリー

検証フェーズ (15 分)

別のチームと認証情報を交換します

使用: 米国東部 (オハイオ)
us-east-2

ワークショップ

https://awssecworkshops.com

https://awssecworkshops.com/workshops/identity-round-robin/permission-boundaries/

[Verify] をクリックしてください

aws

Q & A

プレゼンテーションの質問 2

- リソースを制限せずにパーミッションバウンダリーを実装すると、どのようなリスクがありますか。
- パーミッションバウンダリーとアイデンティティベースポリシーの両方に同じ IAM ポリシーを使用できますか。
- 他のリソースの制限方法より優先されるリソースの制限方法がありますか。