AWS Lambda にも来ました。属性ベースのアクセス制御 (ABAC) サポート

RBACとの比較やユースケース紹介

Kensuke Fukumoto

ISV / SaaS Solution Architect
Amazon Web Service Japan G.K.

自己紹介

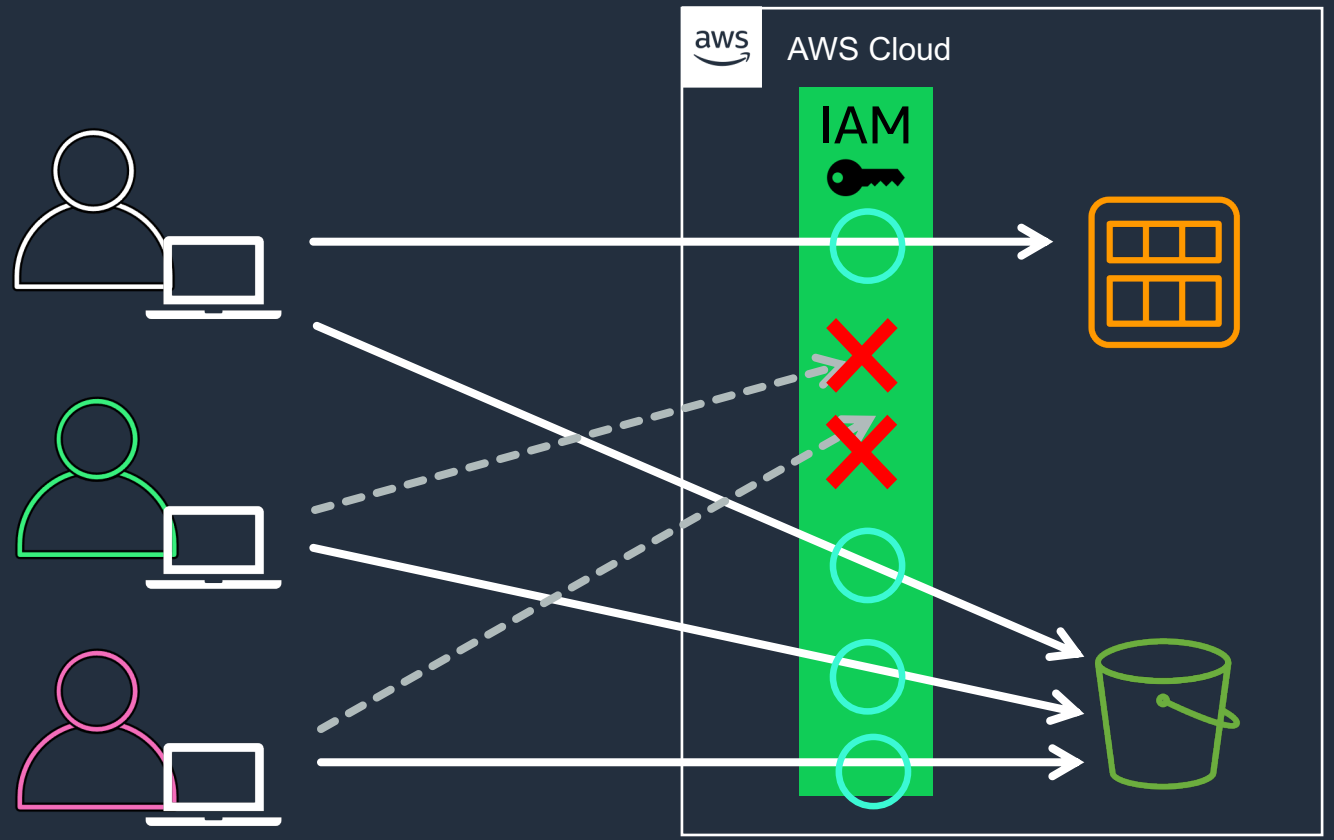
- Solution Architect
- 独立ソフトウェアベンダや SaaSプロバイダのお客様を担当
- 趣味は皇居ラン
- 好きな AWS サービス
  - AWS Lambda
- 関心領域
    - サーバーレスアーキテクチャ
    - SaaS
    - 機械学習

アジェンダ

- IAMとRBAC, ABAC
- LambdaにおけるRBAC,とABAC
- デモ
- まとめ

AWS Identity and Access Management (IAM)

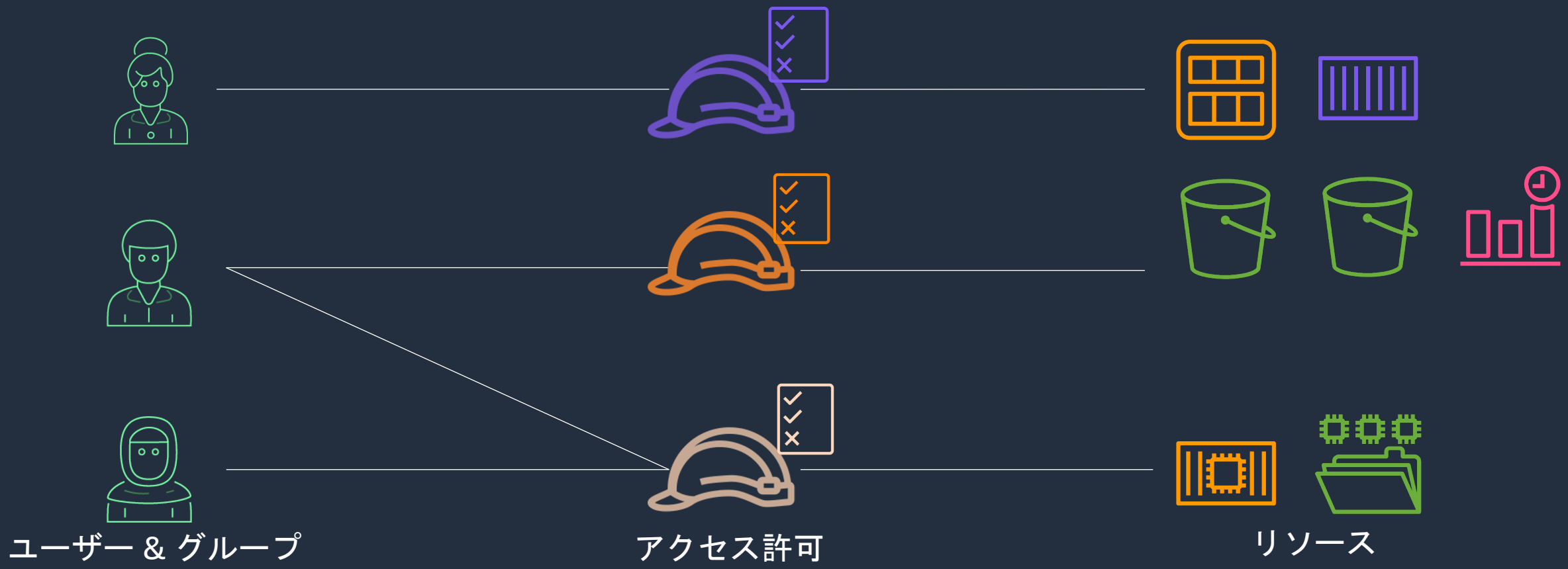
AWS サービスおよびリソースへのアクセスを安全にコントロール



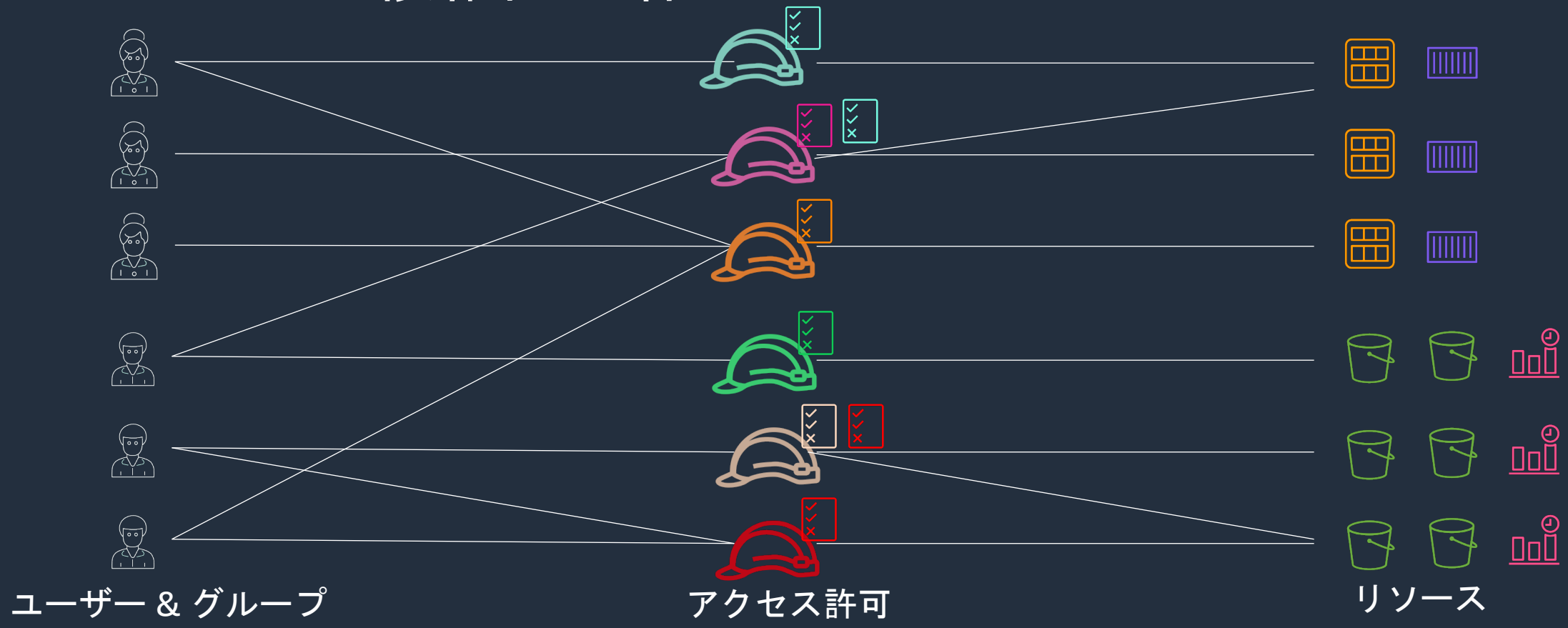
誰が何にどんな条件でアクセスできるかをIAMで制御する。

Role-based access control (RBAC)

リソースにアクセスするための専用の IAM ロールを持つ方式



RBACにおける複雑性の増加



役割やアタッチされたポリシーの複雑さが増すにつれて
複数の IAM ロールを管理することが困難になる場合がある。

スケーラブルな、属性ベースのアクセス許可 (ABAC) モデル

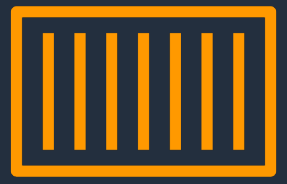
既存のポリシーを更新せずに、属性に基づいてリソースへのアクセスを制御できる。

AWSにおける属性

- AWSにおいて、属性はタグとして管理される。
- タグはキー(必須)と値 (オプション)のペアである。
- タグはコスト配分やリソース検索、アクセス制御などに使用できる。

― 例 ―

project-name = falcon



project-name = falcon
env = dev

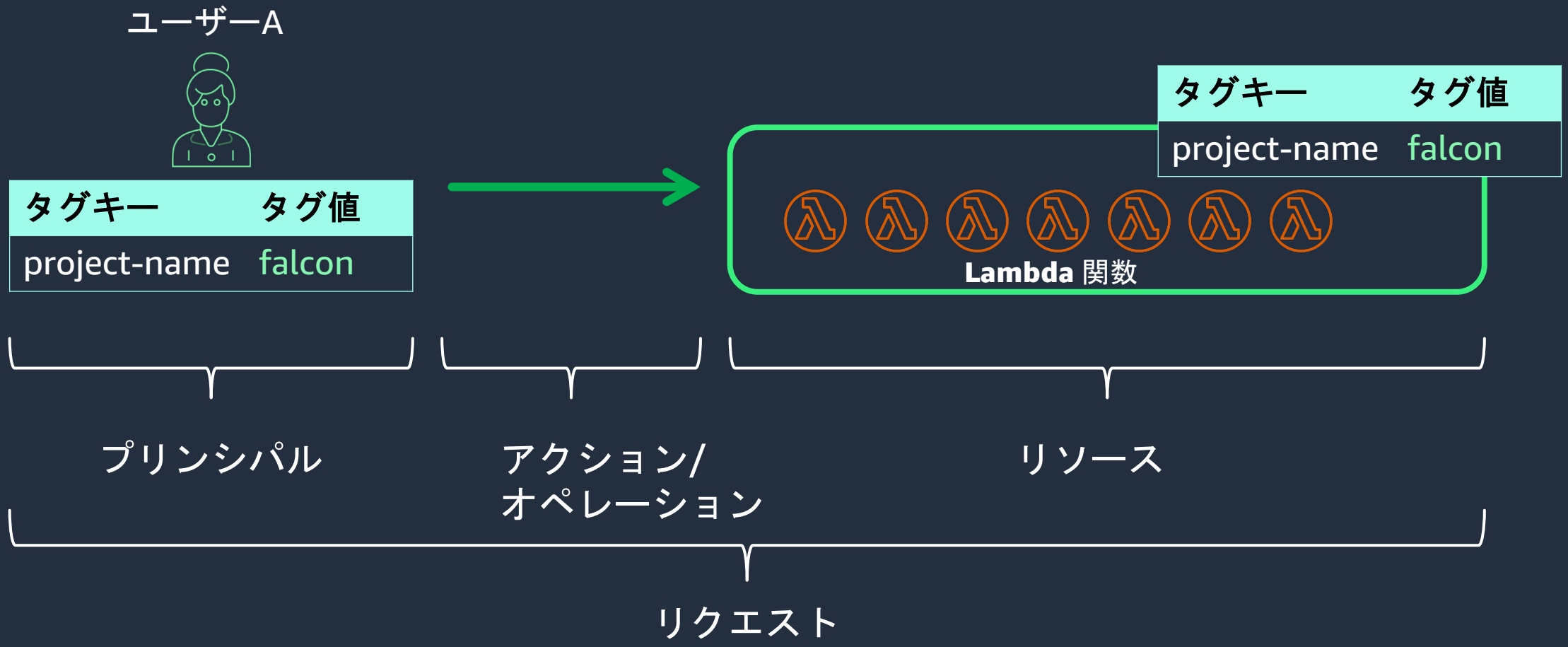
ABACのメリット

- ABAC のアクセス許可は、イノベーションや規制などにおける要件の変化に合わせて柔軟にスケールする。
- 必要なポリシーが少なくなる。
- 属性に基づいて自動でアクセス許可が適用されるためチームは迅速に変化し、成長することができる。
- 新規のユーザやリソースの追加時のアクセス許可の更新なしに、きめ細かなアクセス許可ができる。

そんなABACが
AWS Lambdaにも来ました！

参考ブログ
https://aws.amazon.com/jp/blogs/compute/scaling-aws-lambda-permissions-with-attribute-based-access-control-abac/

用語の整理



Lambdaが対応するグローバル条件コンテキストキー

グローバル条件コンテキストキー: AWSへのリクエスト情報に含まれる
コンテキスト情報の内、特定のサービスに依存しないもの

- `"aws:PrincipalTag/${TagKey}"`

  IAMプリンシパルにアタッチされたタグに基づいてアクセスを許可するかを制御

- `"aws:ResourceTag/${TagKey}"`

  Lambda関数（リソース）にアタッチされたタグに基づいてアクセスを許可するかを制御

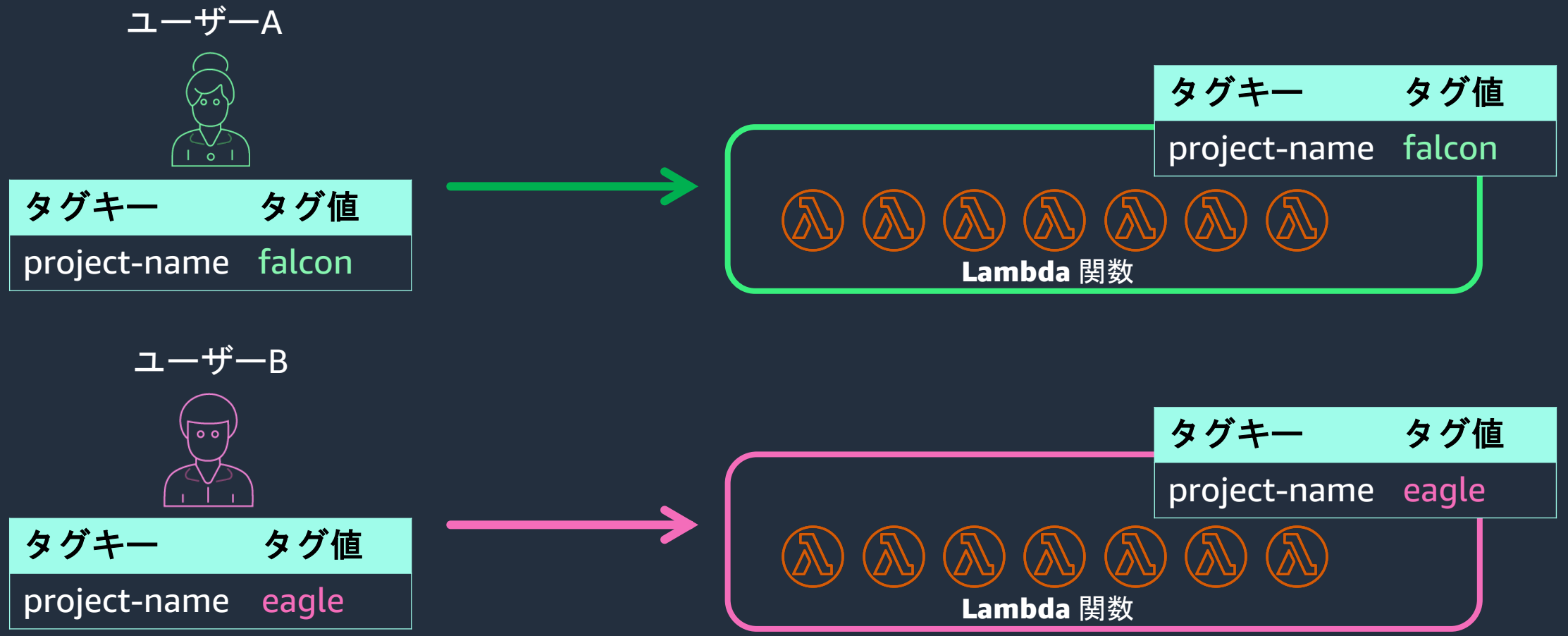
- `"aws:RequestTag/${TagKey}"`

  新規Lambda関数を作成する際などにリクエスト内にタグ付けを要求

- `"aws:TagKeys"`

  リクエスト内で特定のタグが存在することを制御

LambdaにおけるABAC



Lambdaが対応するグローバル条件コンテキストキー

グローバル条件コンテキストキー: AWSへのリクエスト情報に含まれる
コンテキスト情報の内、特定のサービスに依存しないもの

- `"aws:PrincipalTag/${TagKey}"`
  IAMプリンシパルにアタッチされたタグに基づいてアクセスを許可するかを制御
- `"aws:ResourceTag/${TagKey}"`
  Lambda関数（リソース）にアタッチされたタグに基づいてアクセスを許可するかを制御
- `"aws:RequestTag/${TagKey}"`
  新規Lambda関数を作成する際などにリクエスト内にタグ付けを要求
- `"aws:TagKeys"`
  リクエスト内で特定のタグが存在することを制御

LambdaにおけるABAC

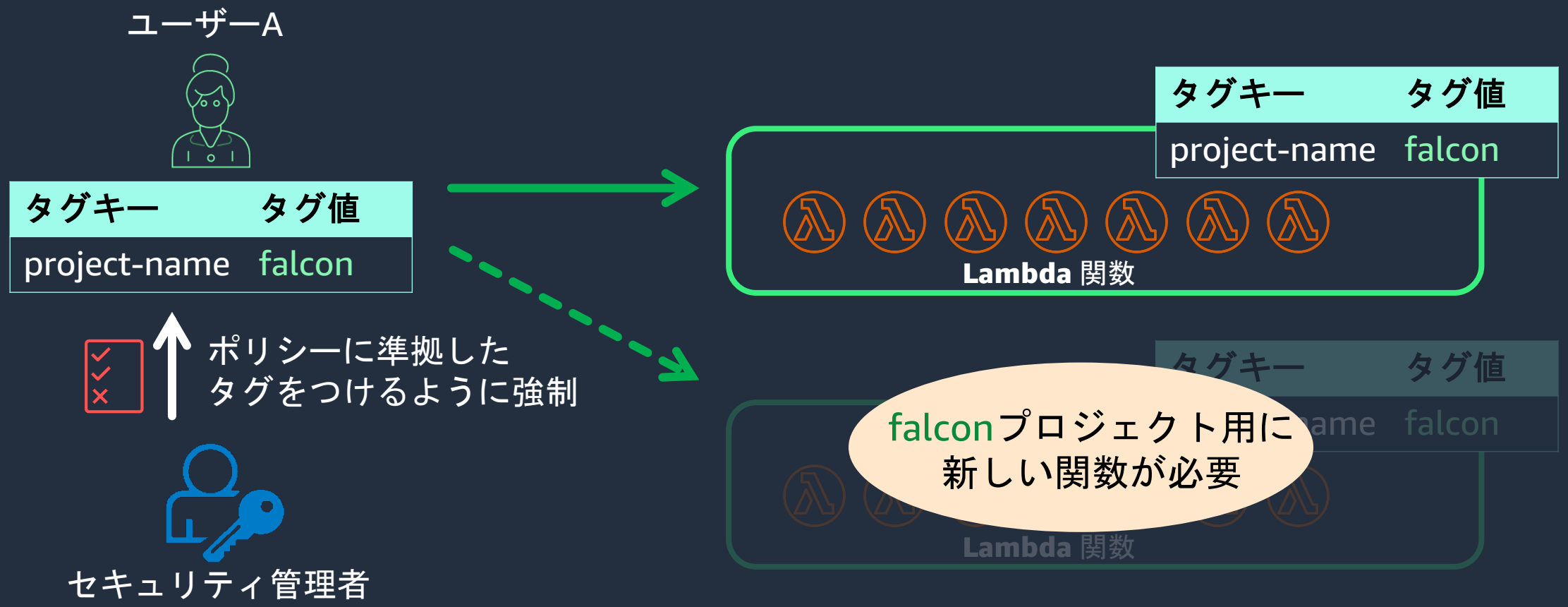
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllActionsLambdaSameProject",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction",
                "lambda:UpdateFunctionConfiguration",
                "lambda:CreateAlias",
                "lambda:DeleteAlias",
                "lambda:DeleteFunction",
                "lambda:DeleteFunctionConcurrency",
                "lambda:GetAlias",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:GetPolicy",
                "lambda:ListAliases",
                "lambda:ListVersionsByFunction",
                "lambda:PublishVersion",
                "lambda:PutFunctionConcurrency",
                "lambda:UpdateAlias",
                "lambda:UpdateFunctionCode"
            ],
            "Resource": "arn:aws:lambda:*:*:function:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/project-name": "${aws:PrincipalTag/project-name}"
                }
            }
        }
    ]
}
```

リソース(Lambda関数) に設定されている
タグ project-name の値がプリンシパルの
タグ project-name の値と同じである事を条件に設定

```
"Condition": {
   "StringEquals": {
      "aws:ResourceTag/project-name":
        "${aws:PrincipalTag/project-name}"
   }
}
```

新規関数作成時のアクセス許可設定

新規に作成されるリソースに対しても適切な形でタグをつける必要がある。



Lambdaが対応するグローバル条件コンテキストキー

グローバル条件コンテキストキー: AWSへのリクエスト情報に含まれる
コンテキスト情報の内、特定のサービスに依存しないもの

- `"aws:PrincipalTag/${TagKey}"`
  IAMプリンシパルにアタッチされたタグに基づいてアクセスを許可するかを制御
- `"aws:ResourceTag/${TagKey}"`
  Lambda関数（リソース）にアタッチされたタグに基づいてアクセスを許可するかを制御
- `"aws:RequestTag/${TagKey}"`
  新規Lambda関数を作成する際などにリクエスト内にタグ付けを要求
- `"aws:TagKeys"`
  リクエスト内で特定のタグが存在することを制御

タグづけを強制するポリシーの設定

`RequestTag` コンディションキーを使用して特定のタグ値をつけるように強制する。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLambdaTagOnCreate",
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:TagResource"
            ],
            "Resource": "arn:aws:lambda:*:*:function:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/project-name": "${aws:Principal
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "project-name"
                    ]
                }
            }
        }
    ]
}
```

キーがproject-nameであるタグの値が
リクエスト者のタグの値になるように強制する。

```
"Condition": {
        "StringEquals": {
            "aws:RequestTag/project-name":
            "${aws:PrincipalTag/project-name}"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": [
                "project-name"
            ]
        }
    }
}
```

キーがproject-nameである
タグ以外は作成できない。

Lambdaが対応するグローバル条件コンテキストキー

グローバル条件コンテキストキー: AWSへのリクエスト情報に含まれる
コンテキスト情報の内、特定のサービスに依存しないもの

- `"aws:PrincipalTag/${TagKey}"`
  IAMプリンシパルにアタッチされたタグに基づいてアクセスを許可するかを制御
- `"aws:ResourceTag/${TagKey}"`
  Lambda関数（リソース）にアタッチされたタグに基づいてアクセスを許可するかを制御
- `"aws:RequestTag/${TagKey}"`
  新規Lambda関数を作成する際などにリクエスト内にタグ付けを要求
- `"aws:TagKeys"`
  リクエスト内で特定のタグが存在することを制御

プロジェクト横断で組織を管理するユーザ

マネージャ

タグキー	タグ値
organization	birds
cost-center	it

セキュリティ管理者

タグキー	タグ値
project-name	falcon
organization	birds
cost-center	it

タグキー	タグ値
project-name	eagle
organization	birds
cost-center	it

マネージャのポリシー設定

ResourceTag コンディションキーを使用してアクセス許可の対象を絞る。

キーが organization、 cost-center であるタグの値が
リクエスト者のタグの値である必要がある。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllActionsLambdaManager",
            "Effect": "Allow",
            "Action": [
                "lambda:GetAlias",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:GetPolicy",
                "lambda:GetPolicy",
                "lambda:ListAliases",
                "lambda:ListVersionsByFunction"
            ],
            "Resource": "arn:aws:lambda:*:*:function:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/organizatio
                    "aws:ResourceTag/cost-ce
                }
            }
        }
    ]
}
```

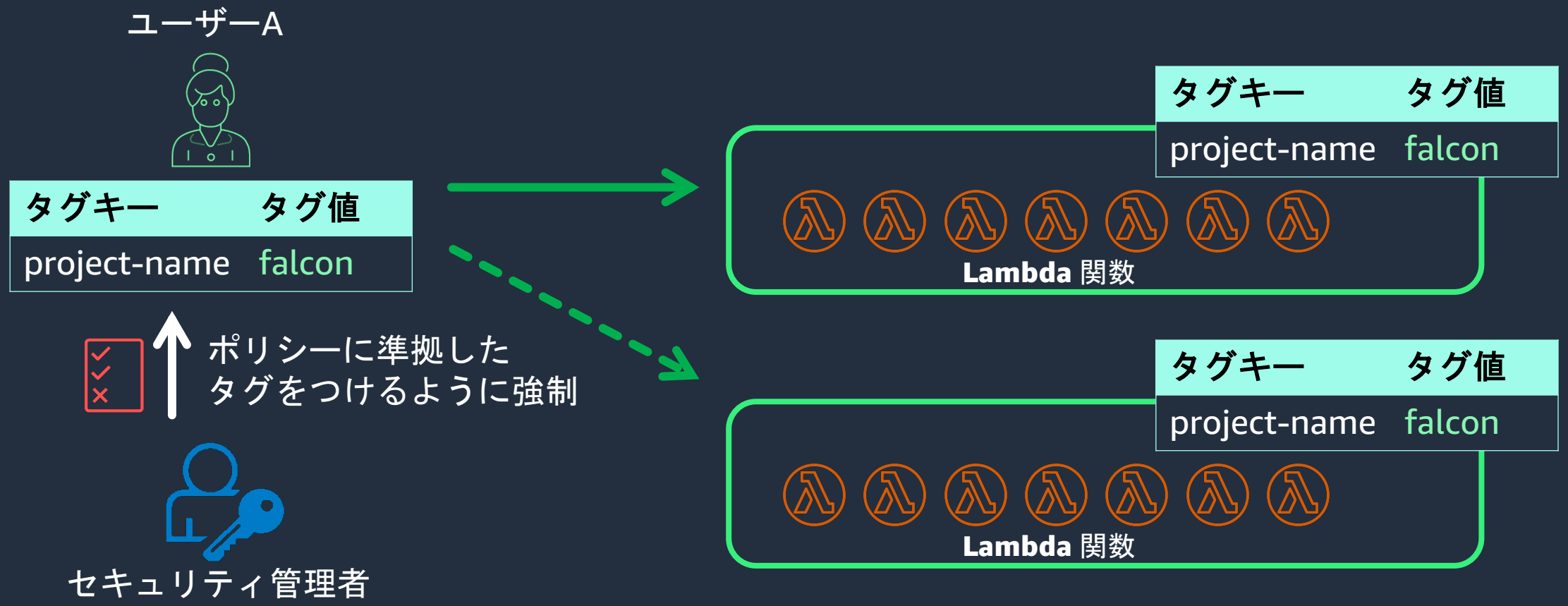
```
"Condition": {
        "StringEquals": {
            “aws:ResourceTag/organization”:
              "${aws:PrincipalTag/organization}",
            “aws:ResourceTag/cost-center”:
              "${aws:PrincipalTag/cost-center}"
        }
    }
```

プロジェクトが属する組織が変更した場合はマネージャにアクセス許可はなくなる。

デモ

新規関数作成時のアクセス許可設定

新規に作成されるリソースに対しても適切な形でタグをつける必要がある。



まとめ

- ユーザの役割が少なく、ポリシーに変化が少ない場合にはRBACが有効

- 役割の変更や追加が頻繁にある場合にはABACによりスケーラブルなアクセス許可を実現可能

- AWS LambdaにもABACが追加され、条件キーを駆使して組織にあったアクセス許可の仕組みを実現

Thank you!