



INNOVATE

ONLINE CONFERENCE

AWSome Day

高橋 敏行

アマゾン ウェブ サービス ジャパン株式会社
技術統括本部 パートナー技術本部
テクニカルイネーブルメント部
パートナーソリューションアーキテクト

平賀 博司

アマゾン ウェブ サービス ジャパン株式会社
トレーニングサービス本部
テクニカルトレーナー

TWITTERハッシュタグ #AWSInnovate

アジェンダ

セッション1

AWS のグローバルインフラストラクチャと
ネットワークおよびコンピューティング

セッション2

ストレージとデータベース

セッション3

AWS のセキュリティの基本

セッション4

Well-Architected Frameworkと料金の話

セッション 3

AWSセキュリティの基本

このセッションでは・・・

次の事柄を学習します。

- ◆ 責任共有モデル
- ◆ VPCのネットワーク制御
 - ・ セキュリティグループ
- ◆ AWSの認証とアクセス管理
 - ・ AWS Identity and Access Management



AWS のセキュリティの紹介

さまざまなAWSのセキュリティ

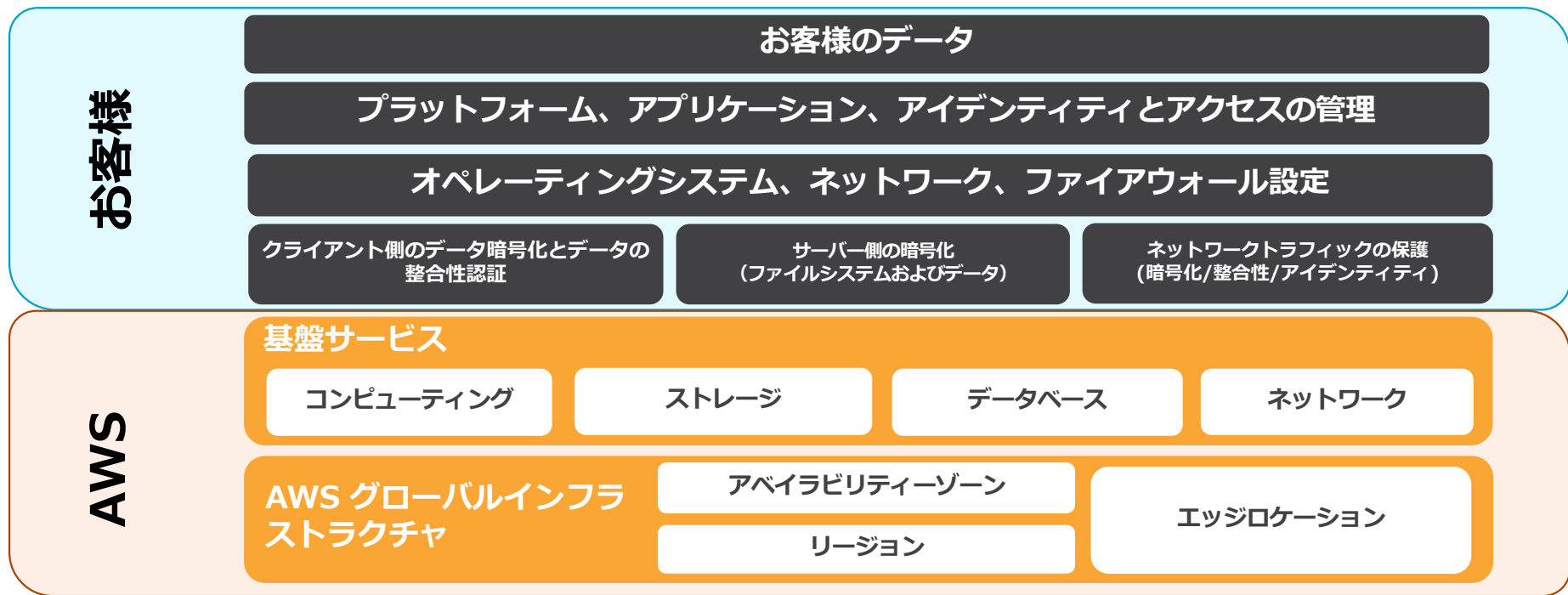
セキュリティは AWS の**最重要事項**です

- ◆ AWSの責任共有モデル
- ◆ ネットワークセキュリティ
 - ◆ セキュリティグループ
- ◆ アクセスコントロールと管理
 - ◆ AWS Identity and Access Management (IAM)
- ◆ データの暗号化
- ◆ モニタリングとロギング
- ◆ セキュリティ製品と機能
- ◆ インベントリと設定の管理

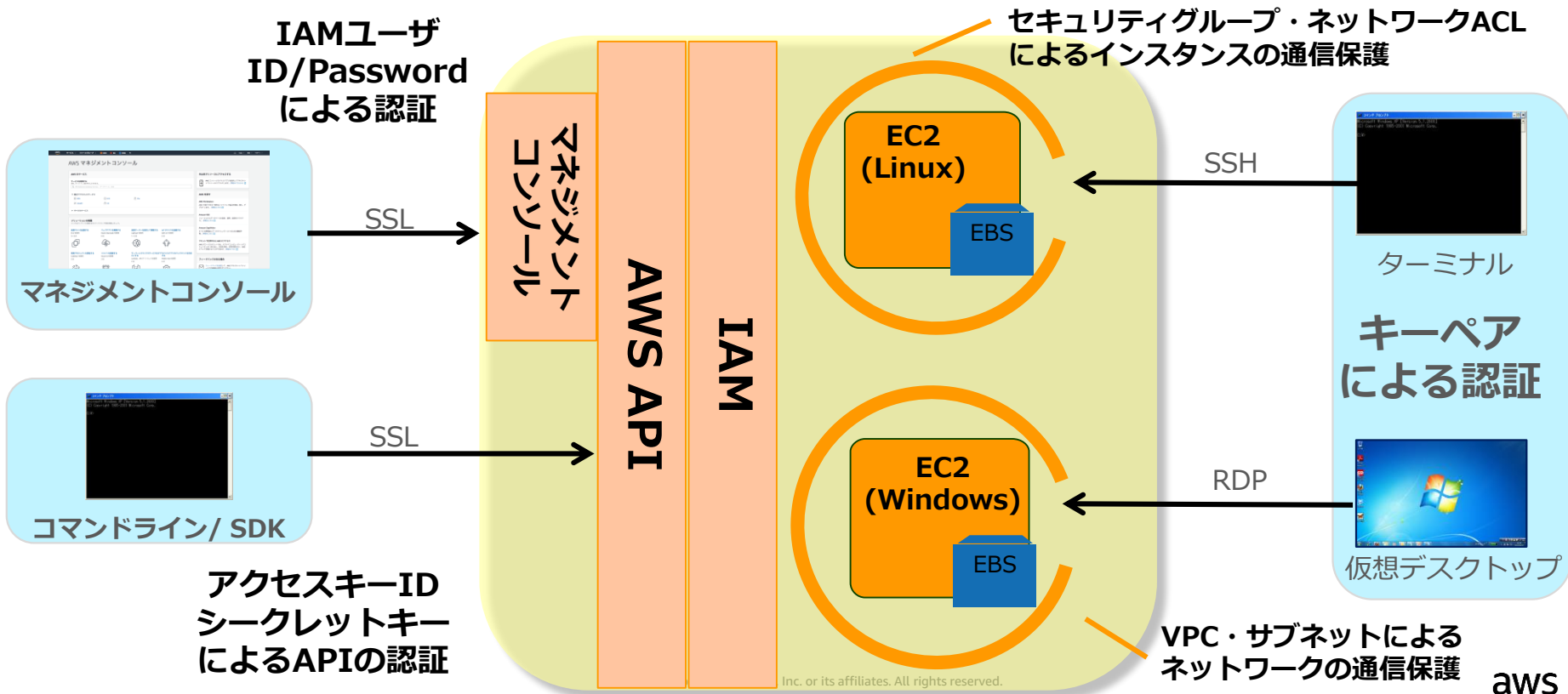


今回

責任共有モデル – AWS



AWS利用におけるセキュリティ確保の全体像



VPCのネットワーク制御

VPCのネットワーク制御機能

◆ ファイアウォール

- 通信を通す・通さないの制御をおこなう
- セキュリティグループ/ネットワークACL

◆ ネットワークの経路制御

- VPC内もしくはVPCの内外の通信経路を設定する
- ルートテーブル

◆ インターネットとの通信

- VPC内とインターネット間で通信が可能
- インターネットゲートウェイ

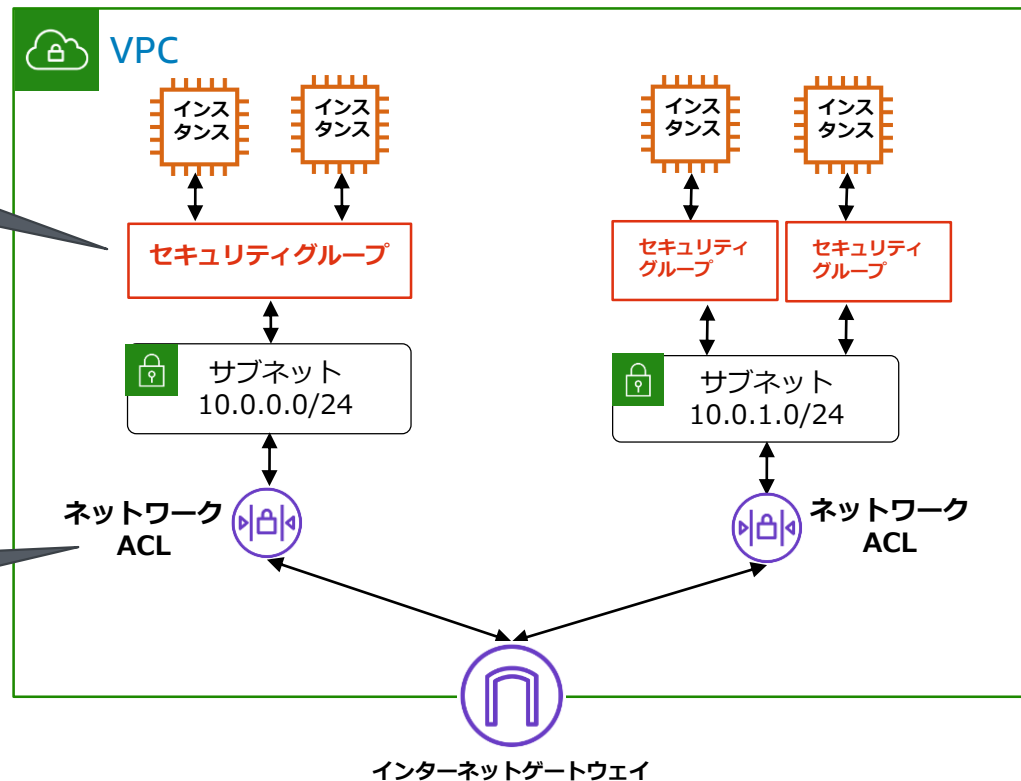
◆ オンプレミス環境との接続

- VPC内とオンプレミス環境間で通信が可能
- Direct Connect/VPN

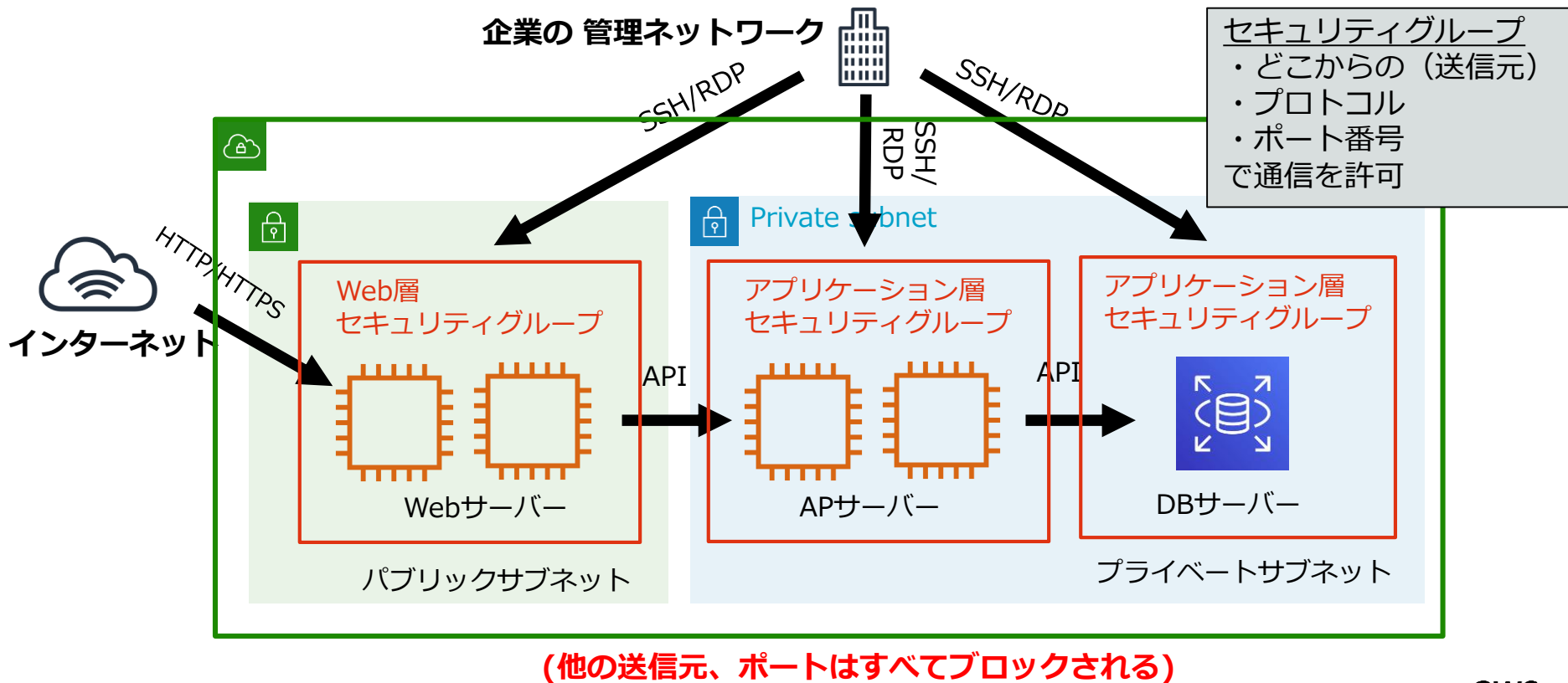
ファイアウォール

インスタンスの
ファイアウォール

サブネットの
ファイアウォール



AWS のセキュリティ



AWS利用におけるセキュリティ確保の全体像

IAMユーザ
ID/Password
による認証

SSL

マネジメントコンソール

SSL

コマンドライン/ SDK

アクセスキーID
シークレットキー
によるAPIの認証

マネジメント
コンソール

AWS API

IAM

セキュリティグループ・ネットワークACL
によるインスタンスの通信保護

EC2
(Linux)

EBS

SSH

ターミナル

キーペア
による認証

RDP

EC2
(Windows)

EBS

仮想デスクトップ

VPC・サブネットによる
ネットワークの通信保護



AWS の認証とアクセス管理

AWS Identity and Access Management (IAM)

◆ AWS リソースへのアクセス制御

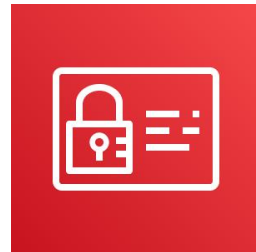
- ・ 認証
- ・ 認可

◆ クラウドサービスへのアクセス

- ・ コンピューティング
- ・ ストレージ
- ・ データベース
- ・ アプリケーションサービス

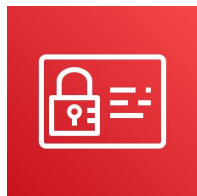
◆ 機能

- ・ ユーザーとそのアクセス権の管理
- ・ ロールとその権限の管理
- ・ フェデレーティッドユーザーとその権限の管理



AWS IAM

- ◆ ユーザーとグループの作成
- ◆ 権限とロールの付与



ユーザー



グループ



権限



ロール

AWS IAM 認証

認証

AWS マネジメントコンソール

ユーザー名とパスワード



IAM ユーザー

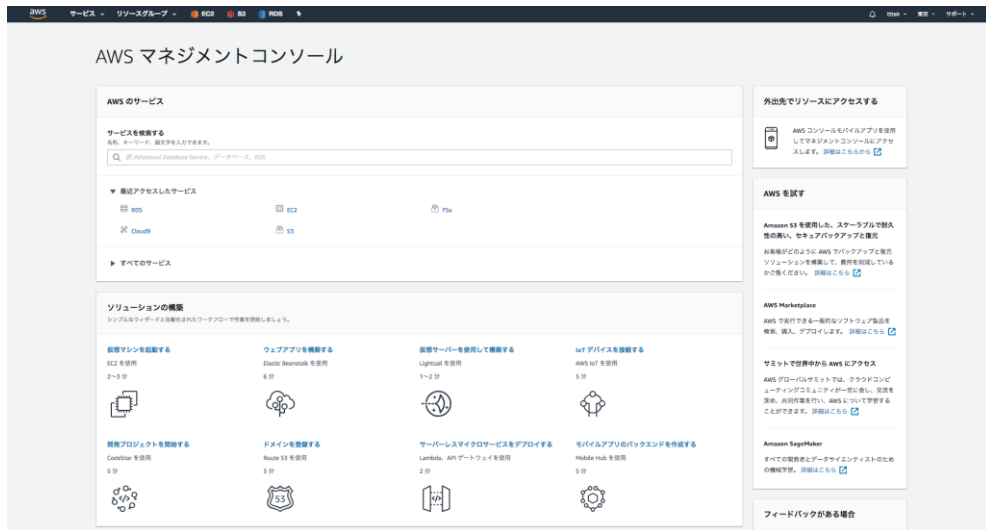
Account:

User Name:

Password:

MFA users, enter your code on the next screen.

Sign In



AWS IAM 認証

認証

AWS CLI または SDK API

アクセスキー ID とシークレットアクセスキー



IAM ユーザー

アクセスキー ID: AKIAIOSFODNN7EXAMPLE
シークレットアクセスキー: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

AWS CLI

```
~ $ aws configure
AWS Access Key ID [*****O22A]:
AWS Secret Access Key [*****4m8i]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

AWS SDK および API



Java



Python



.NET

AWS アカウントルートユーザー

- ◆ アカウントルートユーザーがすべての AWS サービスへの完全なアクセス権を持つ

Create an AWS account

Email address

Password

Confirm password

AWS account name ⓘ

[Continue](#)

[Sign in to an existing AWS account](#)

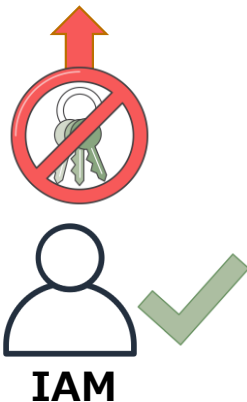
© 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved.
[Privacy Policy](#) | [Terms of Use](#)

AWS アカウントルートユーザー

推奨事項

1. ルートユーザーのアクセスキーを削除する（存在している場合）
2. IAM ユーザーを作成する
3. 管理者アクセス権を付与する
4. IAM 認証情報を使用して
AWS を操作する

newuser@example.com



AWS IAM: 認可

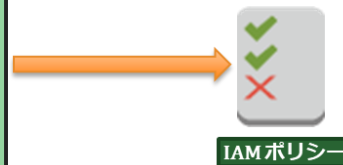
◆ AWS のサービスへのアクセス

- 許可の付与

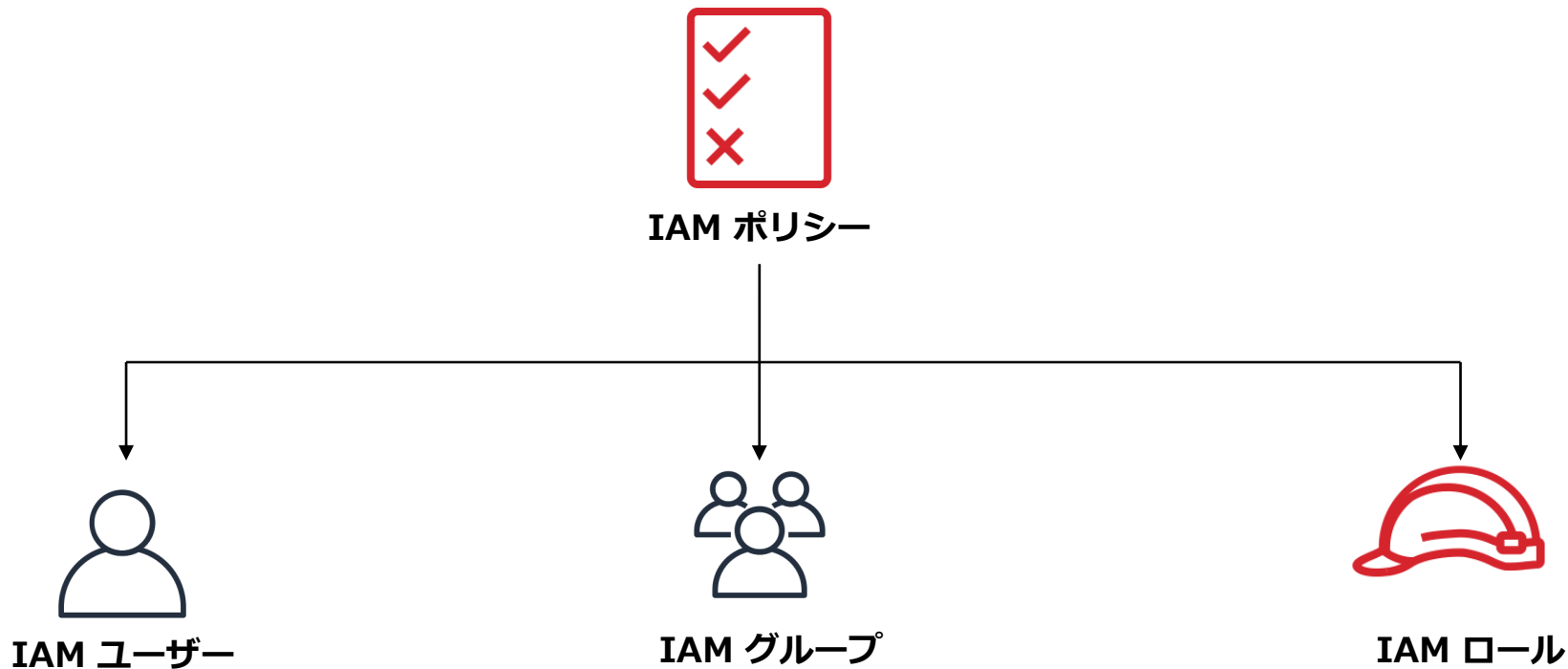
◆ 権限の割り当て

- AWS IAM ポリシーの作成

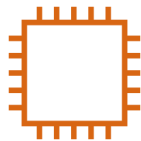
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1453690971587",
      "Action": [
        "ec2:Describe*",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "54.64.34.65/32"
        }
      }
    },
    {
      "Sid": "Stmt1453690998327",
      "Action": [
        "s3:GetObject*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example_bucket*"
    }
  ]
}
```



AWS IAM: ポリシーの割り当て

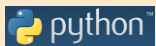


アプリケーションの AWS リソースへのアクセス

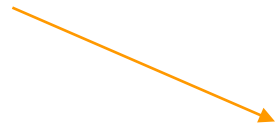


EC2

```
import boto3
client = boto3.client('s3')
response = client.list_buckets()
```

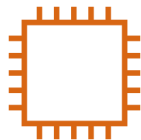


+
アクセスキーID
シークレットアクセスキー



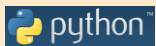
Amazon S3

アプリケーションの AWS リソースへのアクセス



EC2

```
import boto3
client = boto3.client('s3')
response = client.list_buckets()
```



+
アクセスキーID
シークレットアクセスキー



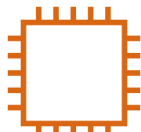
Amazon S3



IAM ポリシー
(S3のアクセス権限)



IAM ロール



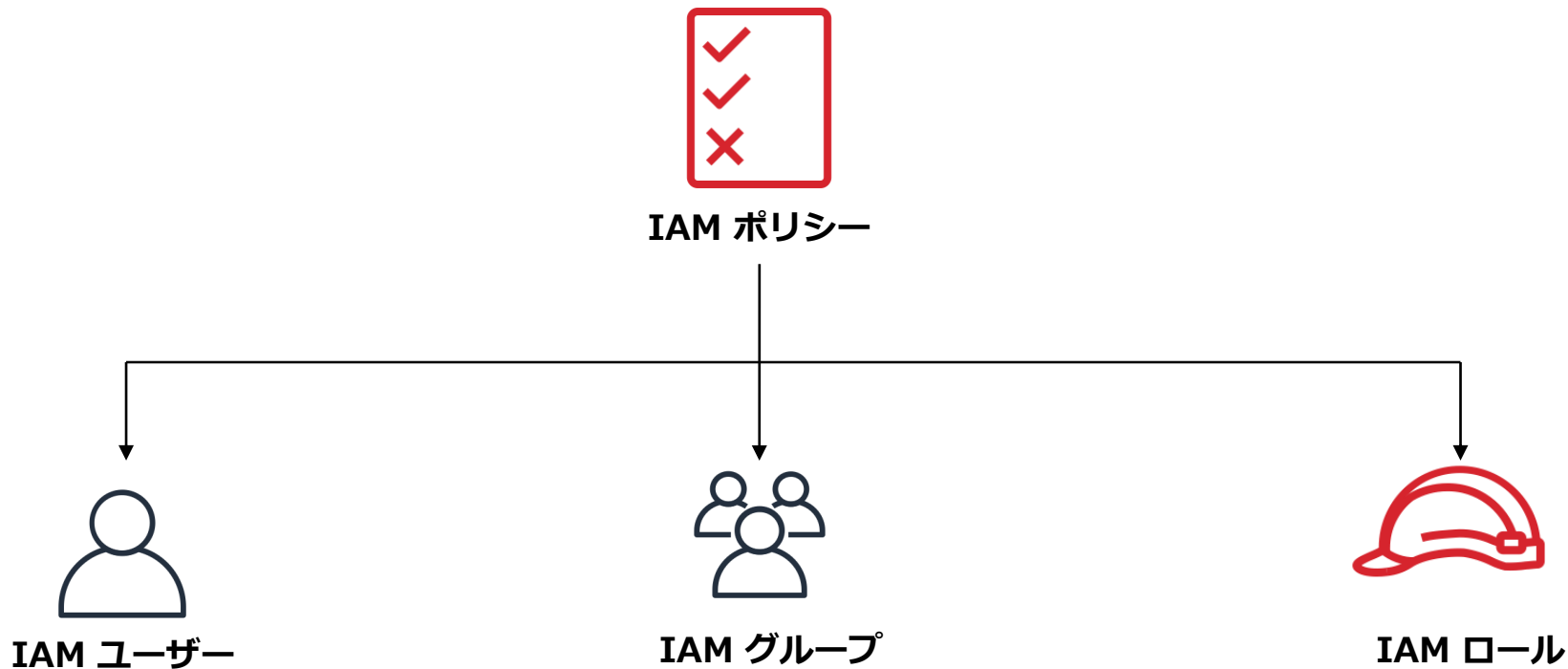
EC2

```
import boto3
client = boto3.client('s3')
response = client.list_buckets()
```

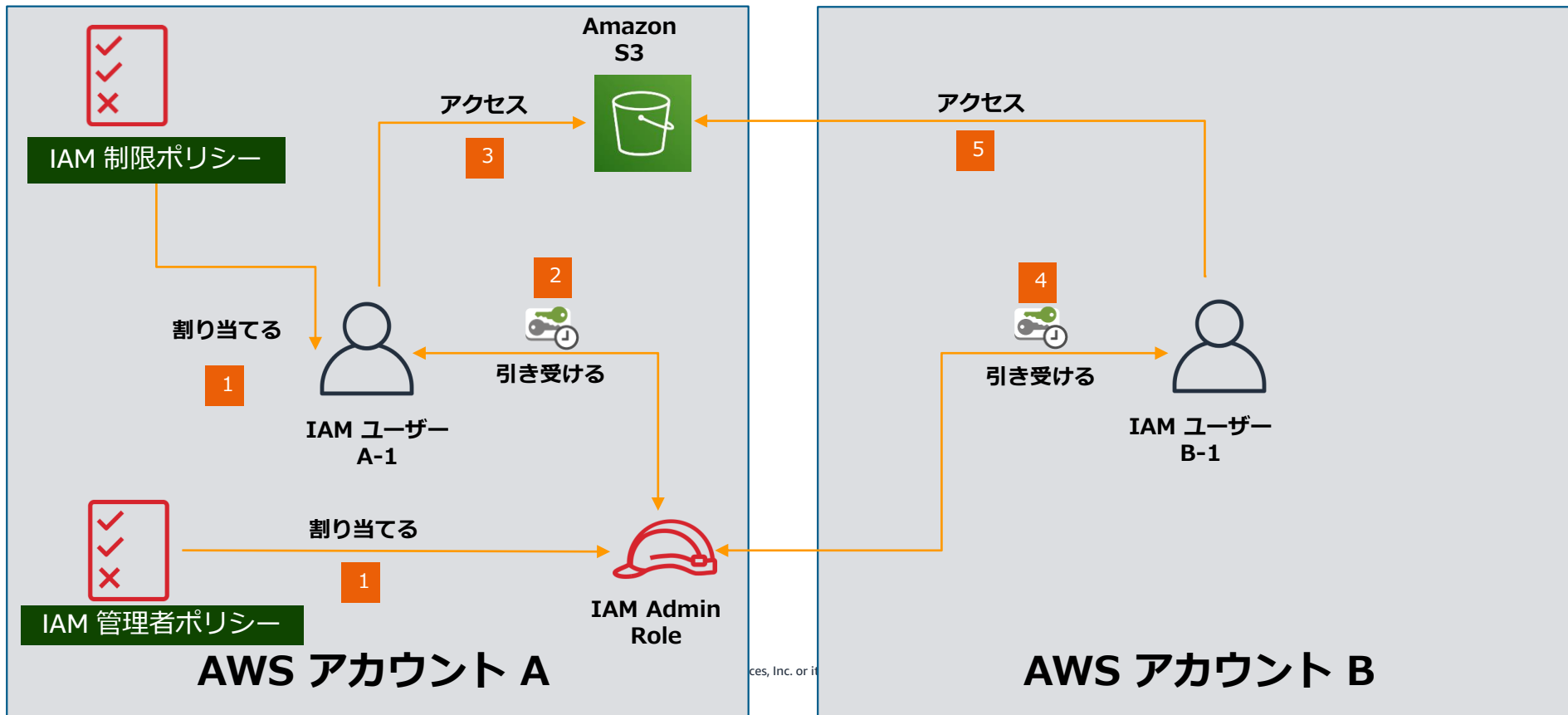


ベストプラクティス

AWS IAM: ポリシーの割り当て



AWS IAM ロール – ロールを引き受ける



AWS IAM のベストプラクティス

- ◆ AWS アカウントの (ルート) アクセスキーがある場合は削除
- ◆ 個々の IAM ユーザーの作成
- ◆ グループを使用して IAM ユーザーに権限を割り当てる
- ◆ 最小権限を付与
- ◆ 強力なパスワードポリシーを設定する
- ◆ 権限のあるユーザーに対して MFA を有効にする



AWS IAM のベストプラクティス (続き)

- ◆ Amazon EC2 インスタンスで実行されるアプリケーションに対して IAM ロールを使用する
- ◆ 認証情報を共有するのではなく、
ロールを使用して委任する
- ◆ 定期的に認証情報を更新する
- ◆ 不要なユーザーと認証情報を削除する
- ◆ 追加のセキュリティのために
条件(Condition)を使用する
- ◆ AWS アカウントのアクティビティを
モニタリングする

このセッションで学んだこと

次の事柄を学習しました。

- ◆ 責任共有モデル
- ◆ VPCのネットワーク制御
 - セキュリティグループ
- ◆ AWSの認証とアクセス管理
 - AWS Identity and Access Management



アジェンダ

セッション1

AWS のグローバルインフラストラクチャと
ネットワークおよびコンピューティング

セッション2

ストレージとデータベース

セッション3

AWS のセキュリティの基本

セッション4

Well-Architected Frameworkと料金の話

Thank you!

高橋 敏行
平賀博司

内容についての注意点

- 本資料では2019年3月時点でのサービス内容および価格に基づいたスライドや説明になっています。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。
- 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.