

# SECKUN:クラウドセキュリティ（第二回）

Amazon Web Services Japan

Head of Security Assurance, Japan

松本照吾（matshogo@）



# Design for Failure with AWS

Tools to make your life easier









# 前回の振り返り

- クラウドによってITがどうかわったか



Mentimeter

## 今日はここまで

- 本日はあまり"セキュリティ"を出さずに、その背景を伝えてきました。
- クラウドならではの技術や設計の考え方の延長に"クラウドセキュリティ"が存在します。
- 次回はクラウドによるセキュリティを中心にお伝えします

3



今回も：サービスの細かい話などはあまりしません  
あと、感想、質問はSlackに（質問は 質問：って書いてもらえると嬉しいです）



100を超えるAWSアカウント運用におけるガードレール構築事例

AWS JAPAN SUMMIT ONLINE 2020

2020/10/15

**100を超えるAWSアカウント運用におけるガードレール構築事例**

先日行われました AWS JAPAN SUMMIT ONLINE 2020 にて、「大規模な組織変遷と100以上のAWSアカウントの横断的セキュリティ...

 長原 佑紀

<https://engineering.visional.inc/blog/>



**aws Security Roadshow Japan**

[今すぐ申し込む >>](#)

### イベント概要

「AWS Security Roadshow」は、クラウドのセキュリティ・コンプライアンスに関する最新の情報をお客様にお届けすることを目的としています。


現在、多くの企業や組織ではテレワーク、リモートワークに移行しており、「ゼロトラスト」に対する議論の活発化等、セキュリティモデルも大きな変革の時期を迎えています。

本イベントでは、お客様が抱える様々なセキュリティに対する課題とその対応方法、実際のお客様のAWSのセキュリティサービス活用事例、今後必要となるセキュリティ対策、AWSのセキュリティの方向性などを包括的にご説明する予定です。

企業・組織の中で、セキュリティやコンプライアンスに課題をお持ちの皆様（営業、マーケティング、開発や運用など）、セキュリティに関する意思決定に関わる管理職や役員の方、公共機関や金融機関のセキュリティ担当者など、AWSセキュリティの最新情報を知りたい方は是非ご参加ください。

日時： 2020年10月28日(水) 10:00-16:30  
会場： オンライン開催  
参加費： 無料（要事前申込）  
定員： 500名

※お申し込みが多数の場合には、定員になり次第締め切らせていただきます。

<https://aws.amazon.com/jp/about-aws/events/2020/securityroadshow2020/> 

# DevSecOpsは“文化”、これってなんだろう

- 最近、DevOpsやゼロトラストでも“単にツールではない”というメッセージは多い
- 文化ってすごいざっくり
- 変化する中における“文化”の維持や促進って





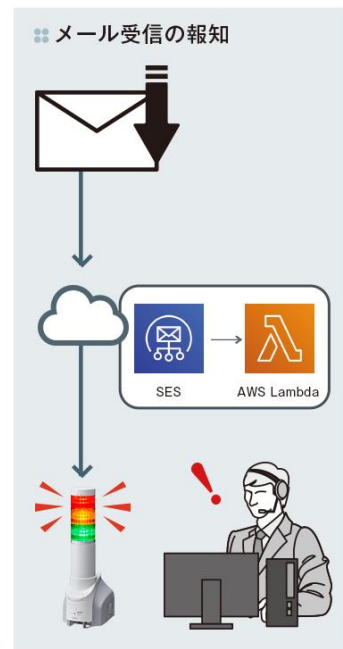
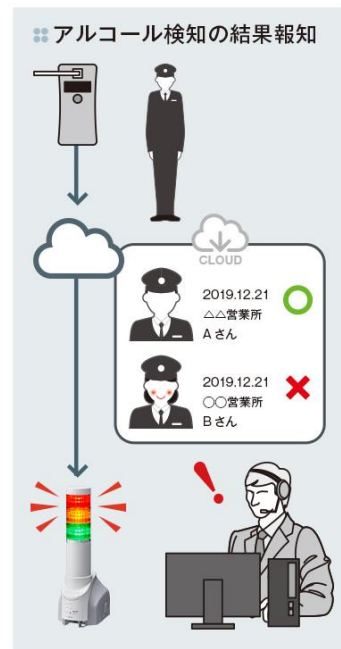
# うちの業務はクラウドとは関係ない？



powered by  
**aws**

対応機種：  
NH-FV シリーズ

UL Φ25 Φ40  
Φ60 音圧 88dB MP3



[https://www.patlite.co.jp/lp/aws\\_integration/](https://www.patlite.co.jp/lp/aws_integration/)



# アイスブレイク：前回講義やラボでの気づきシェア

まずは個人でひとりブレスト（書き出してみる）

ー＞ つぎにグループで意見を出し合う

ー＞ 出た意見を自分でまとめてみる。後でアンケートにかけるように



**本日、考えていきたいこと：  
クラウドがセキュリティにもたらしたもの**

**CISOやエンジニアはクラウドセキュリティにどのように向き合うのか**

# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ



# ITをとりまく変化

**サービスをよりニーズにあった形で提供しつづけることが目的**

- 競争の加速化
  - ニーズを踏まえ、タイミングよくサービスを投入することへの期待
- サービスへの読めない需要
  - デバイスの増加
  - ネットワーク帯域の向上
- 技術の変化
  - IoT、ビッグデータ
  - クラウド
  - AI





反社会人サークル @hanshakaijin · 2018年4月25日

【ゲムマ1日目I36】 #反社会人サークル 新作です！タイトルは『顧客が本当に必要だったものゲーム』。イベント価格1500円。リンク先にて取置予約も実施中。内容はタイトルからご察してください。ようこそ！混迷のシステム開発現場へ——。 #ゲムマ #ゲムマ2018春 #顧客ゲーム [bit.ly/kokyakug](https://bit.ly/kokyakug)



<https://twitter.com/hanshakaijin/status/988911524415590400>



# 変化におけるやっかいなもの：セキュリティ

- セキュリティもニーズの一つ
- ビジネス価値の定量化が困難
- 対応すべきセキュリティは変化
- 当初想定していなかった対応も
- 対応自体がサービスを阻害することもある



# クラウドってなんでしたっけ？

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

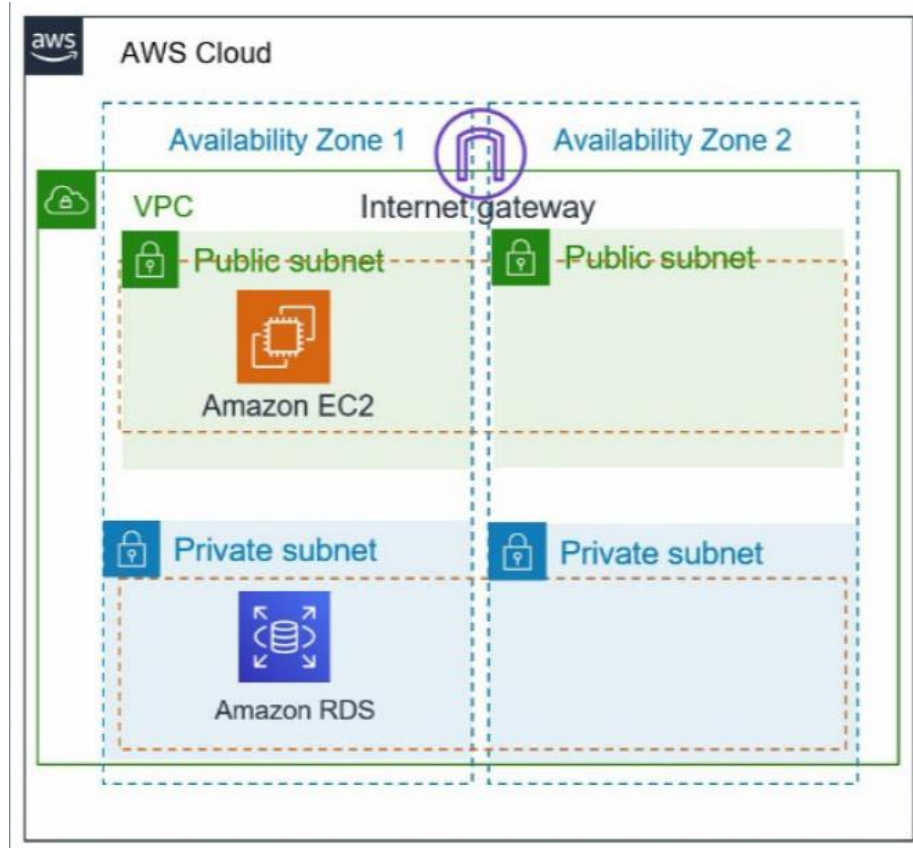




# AGENDA

- ラボ 1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ 2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ 3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ

# ラボ 1 と前回の講義から



# こちら、おすすめ“AWS Hands-on for Beginners”

- AWSJが提供するハンズオン環境解説動画付き（講師もコンテンツも日本語）
- AWSの無料枠をうまく活用（完全無料を確約するものではないですが）

## AWS ハンズオン資料

AWS をステップバイステップでお試しいただくのに役立つ動画および資料を掲載しています。

その他の資料は以下をご覧ください。

初心者向けの資料 »

サービス別の資料 »

[AWS オンラインセミナースケジュール »](#)  
[AWS クラウドサービス活用資料集トップ »](#)

AWS Hands-on for Beginners - ハンズオンははじめの一步: AWS アカウントの作り方 & IAM 基本のキ

[概要](#) [サンプルビデオ](#) [アジェンダ](#) [Speaker](#)

## なぜ、今回はQwiklab ?

- ニーズに即した環境準備の手間
- セキュリティ管理
- コスト管理



視聴はこちら

“AWS Hands-on for Beginners - ハンズオンははじめの一步” 編では、AWS アカウントの作成と IAM に関する説明 / ハンズオンを行います。「AWS をこれから利用していきたい！」という方お話しする機会がよくあるのですが、最初の部分、つまりアカウント作成の部分で苦戦されている方がよくいらっしゃいます。このハンズオンを通して、スムーズにはじめの一步を踏み出していただくのがこのハンズオンのひとつ目のゴールです。また、AWS の各サービスやリソースへのアクセスを安全に管理するための AWS Identity and Access Management (IAM) というサービスがあります。こちらについても「実はよく分かっていない。」という相談をよくいただきます。本ハンズオンシリーズの後半で、IAM ユーザーや IAM ポリシーといった機能の説明を行い、その後、実際に実験を通して IAM の基本を学んでいただくことがもうひとつのゴールになります。

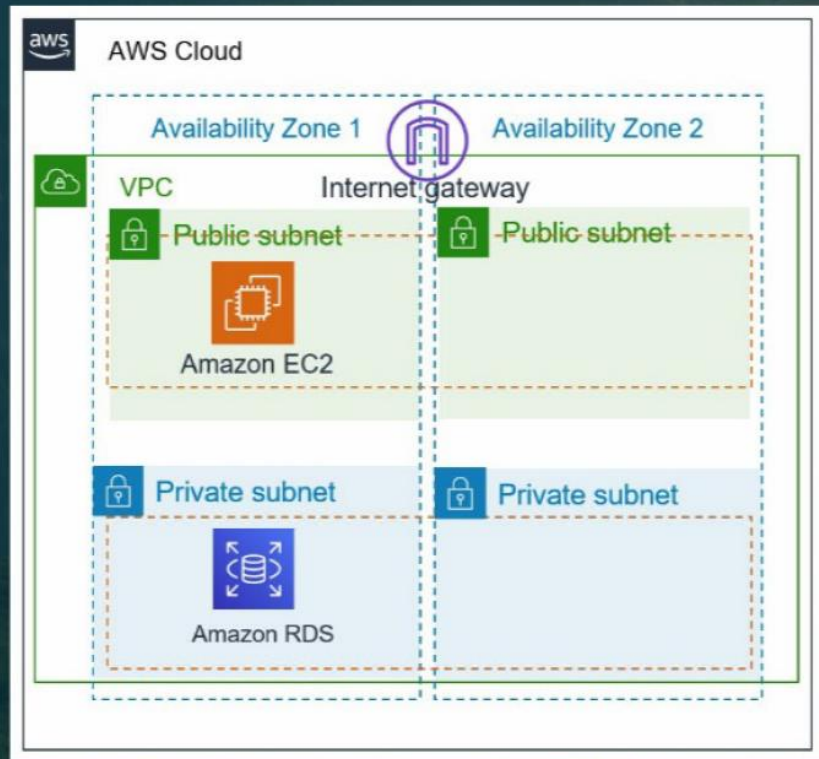
### このような方におすすめ

- AWS を使い始めたいけど、何から始めていいの？という方
- アカウントを作って他のハンズオンをやりたい！という方
- IAM についてサクッと基本を抑えたい！という方

[https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/?trk=aws\\_blog](https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/?trk=aws_blog)



# ラボの振り返り：ラボ 1



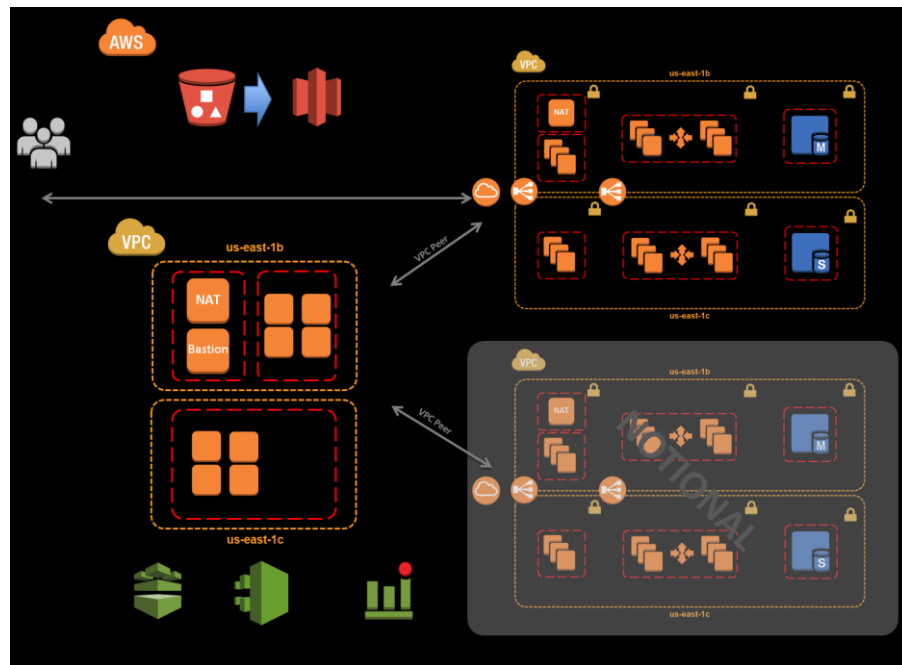
## ラボ（事前学習）の振り返り

- VPC（仮想ネットワーク）内のWebアプリケーション
- フロントにEC2によるWebサーバ
- バックエンドにRDSによるMySQL



# ラボ、手作業大変ですよね

- どこで間違ったのか。。。
  - ログがあれば分析できるかも。
  - ミスするようなところはアラート出してくれればいいのに。
- 環境設定のエラー
  - “人”という脆弱性
  - 誰でもができる（スキルのない人も扱う）
- そもそも手作業の数が少なければミスもないかも



# EC2とRDS、何が違いました？

マネージドサービスとそうではないサービスの違い

操作するレイヤが違った。

汎用コンピュータとDBの違い

OSが選択できるできないか

EC2は何でもソフトウェアをインストールできるけど、RDSはデータベースに特化したマネージドサービス（バックアップとかもある）

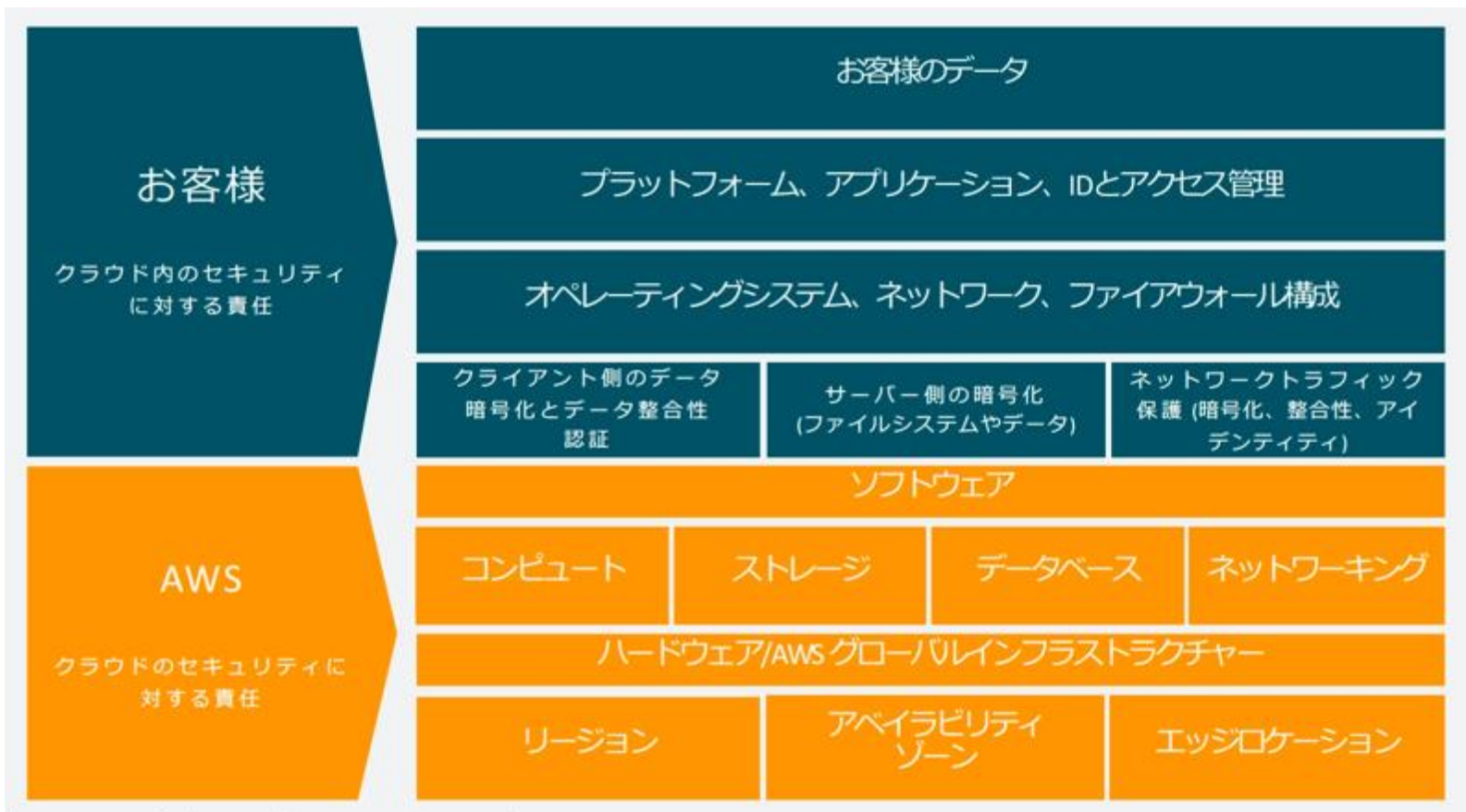
カスタマイズ性？？

PaaSとSaaS

OSのレイヤーから選択するか、アプリから選択するかの違い

EC2はOSが選べ、構築したOS上に自由にミドルウェアがインストールできるが、RDSはOSを意識することなく、データベースを利用することができる。

# 責任共有モデル



# 選択するサービスによって“責任範囲”が異なる

アプリケーション作成	アプリケーション作成	アプリケーション作成	アプリケーション作成
スケールアウト設計	スケールアウト設計	スケールアウト設計	スケールアウト設計
定形運用設計	定形運用設計	定形運用設計	定形運用設計
ミドルウェアパッチ	ミドルウェアパッチ	ミドルウェアパッチ	ミドルウェアパッチ
ミドルウェア導入	ミドルウェア導入	ミドルウェア導入	ミドルウェア導入
OSパッチ	OSパッチ	OSパッチ	OSパッチ
OS導入	OS導入	OS導入	OS導入
HWメンテナンス	HWメンテナンス	HWメンテナンス	HWメンテナンス
ラッキング	ラッキング	ラッキング	ラッキング
電源・ネットワーク	電源・ネットワーク	電源・ネットワーク	電源・ネットワーク
オンプレミス	独自構築 on EC2	マネージドサービス	サーバーレス アーキテクチャ
開発者が担当	AWSが担当		



# IaaS,PaaS,SaaS : 何をサービスとして提供するか

アプリケーション

アプリケーション

アプリケーション

アプリケーション

ミドルウェア

ミドルウェア

ミドルウェア

ミドルウェア

インフラ

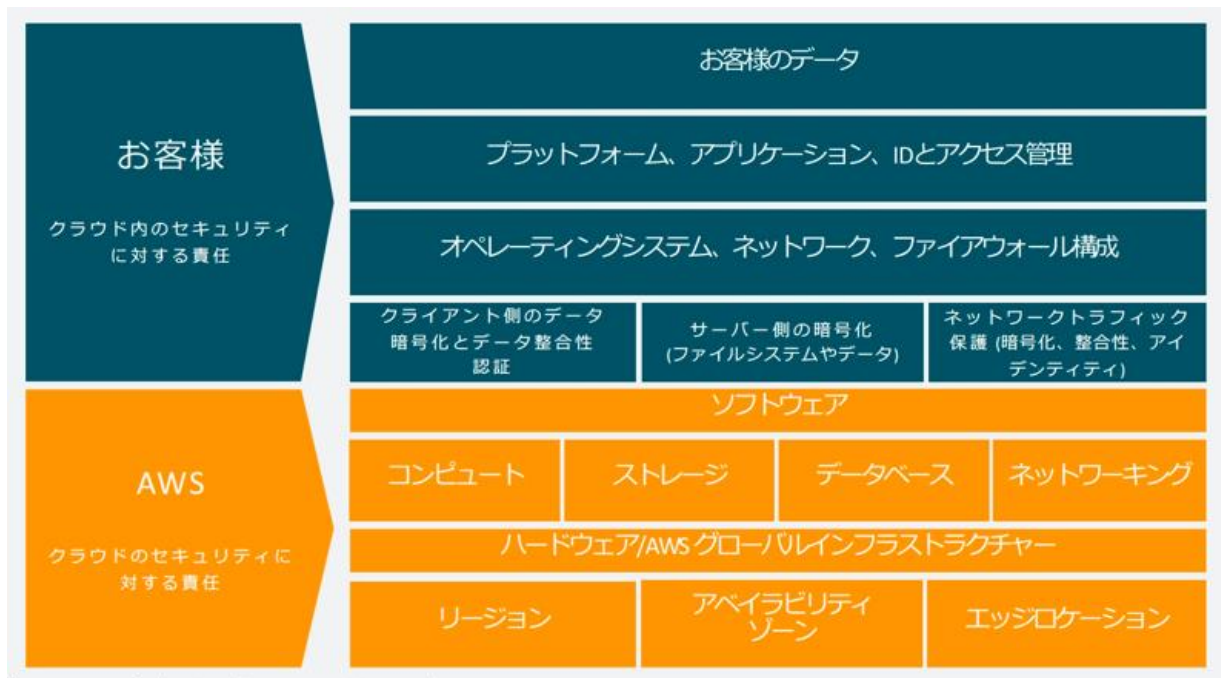
インフラ

インフラ

インフラ

# Security in the Cloud, of the Cloud, そしてBy the Cloud

取り入れることによるAgilityの獲得？ 自社構築、運用（組織やポリシー改訂）？



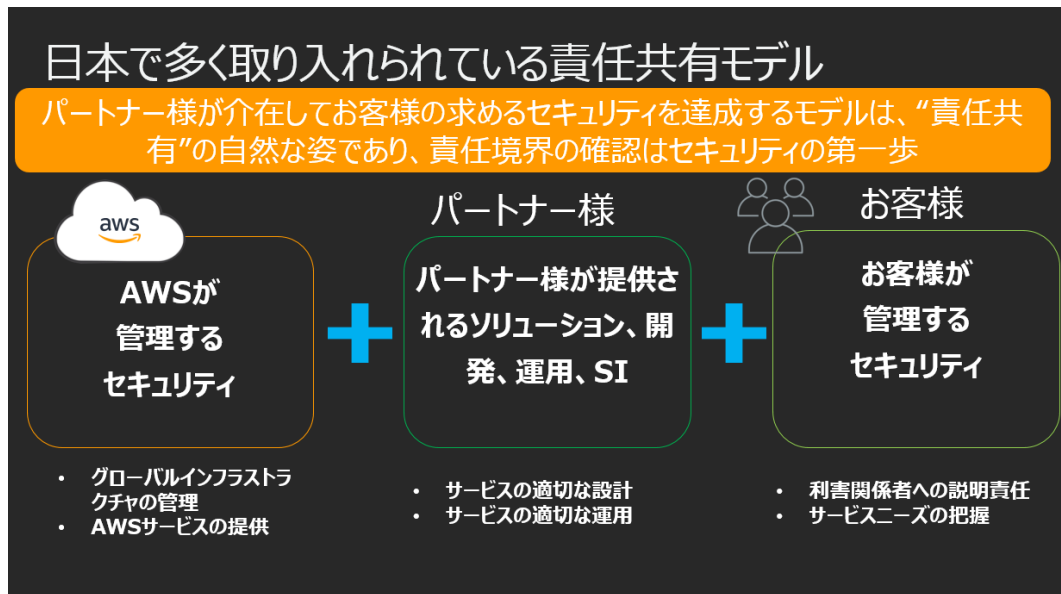
Security系  
SaaS等



外部のサービスはクラウド化  
ID管理、認証基盤  
脅威分析、ログ分析  
WAF



# 管理主体によって“責任範囲が変わる”



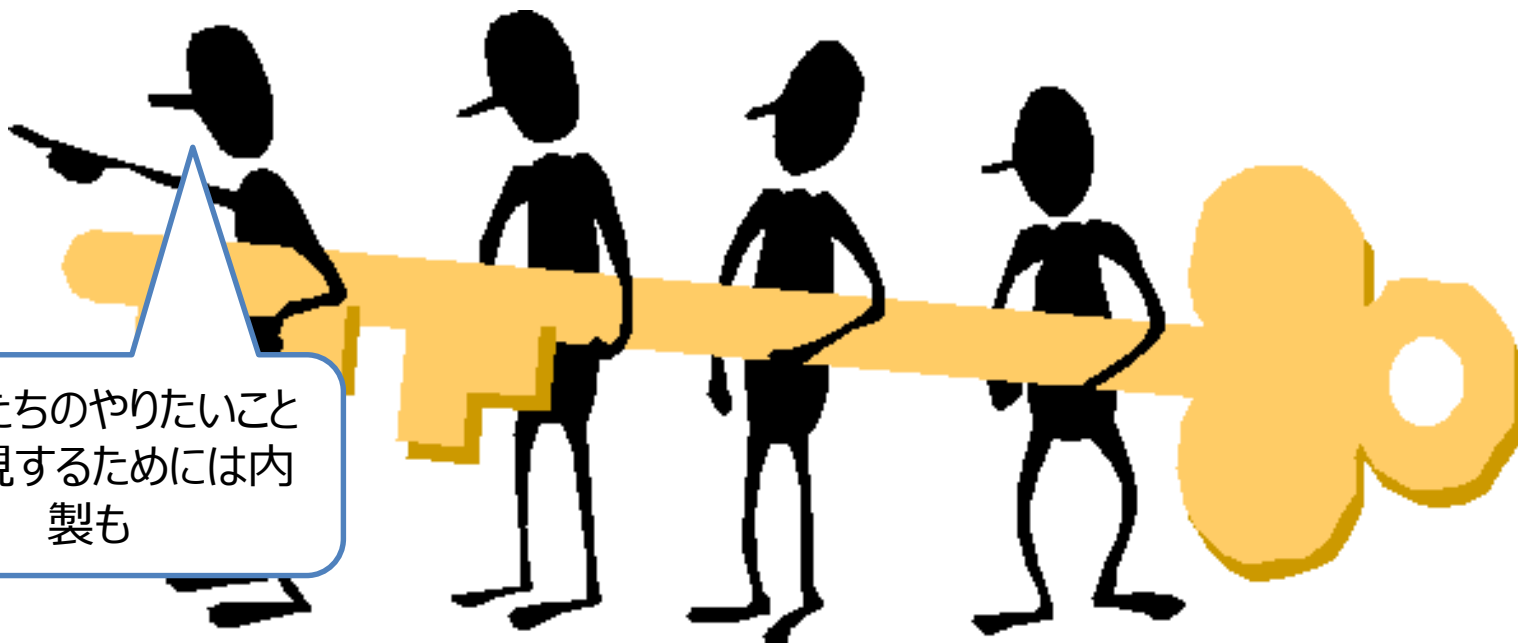
何を提供し、何に責任を持っているのか？

サービス：サービスの構成要素に関しての、説明責任は限定的（Service provider）

SI：サービスの構成要素に関しても、説明責任を有する。（Integrator）



# “利用者”の中でも責任範囲が変わる



自分たちのやりたいことを実現するためには内製も

ビジネス部門  
サービスオーナー  
アクセス設計

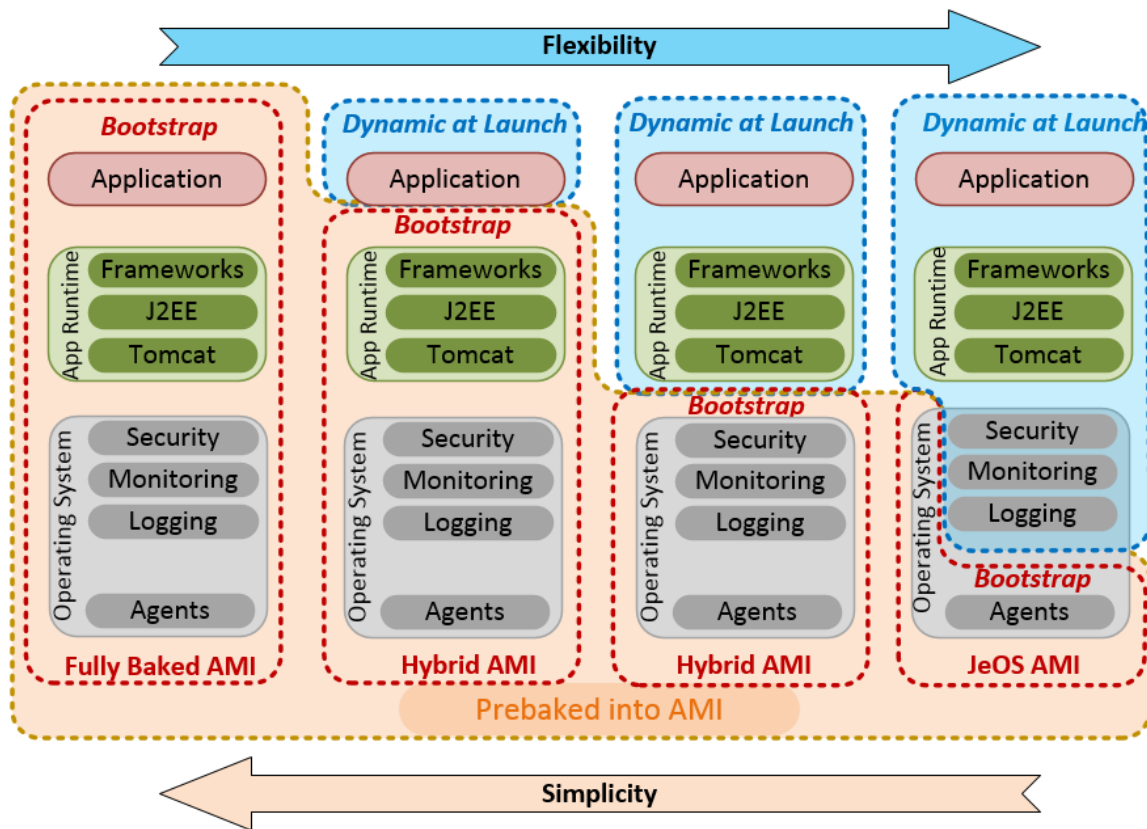
システム部門  
設計・運用

購買  
サプライヤ管理

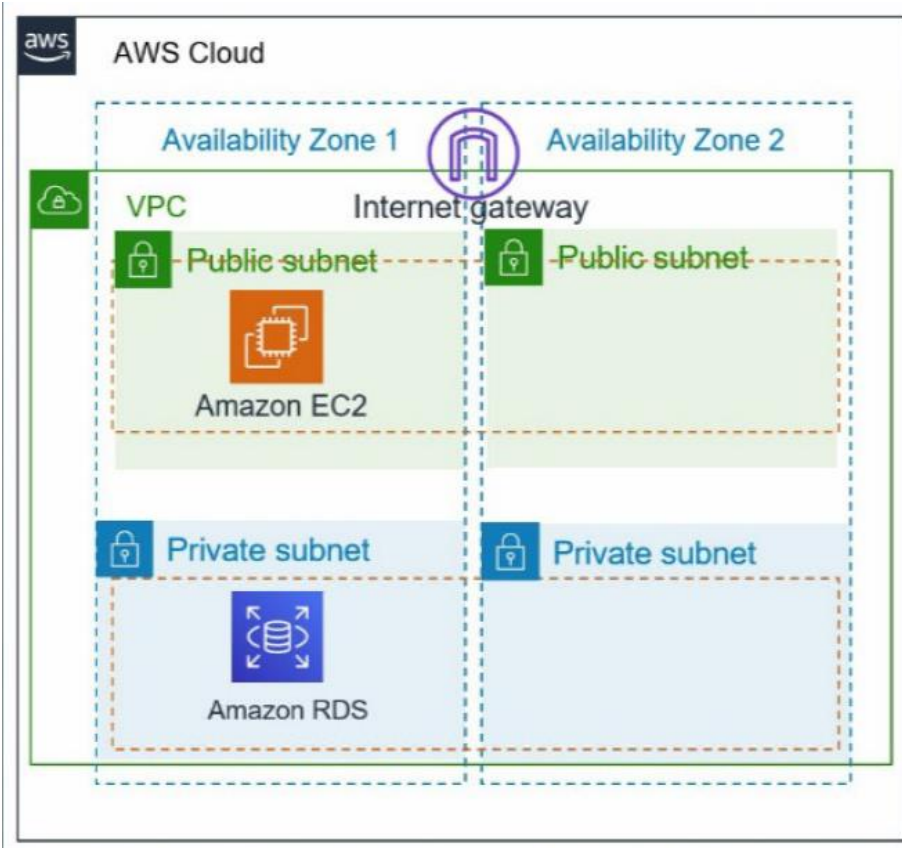
CSIRT  
セキュリティ監視運用



# サーバー一つでも責任共有が発生する



# ラボ1：アーキテクチャの課題



- 大量のトラフィック（DDoS等）に耐えられるか？
- Web ServerがPublic Subnetに存在する（SSH、直接公開？）
- Availability Zone 1 に障害が発生したらサービス停止（DBのフェイルオーバーやEC2のスケールを考慮）

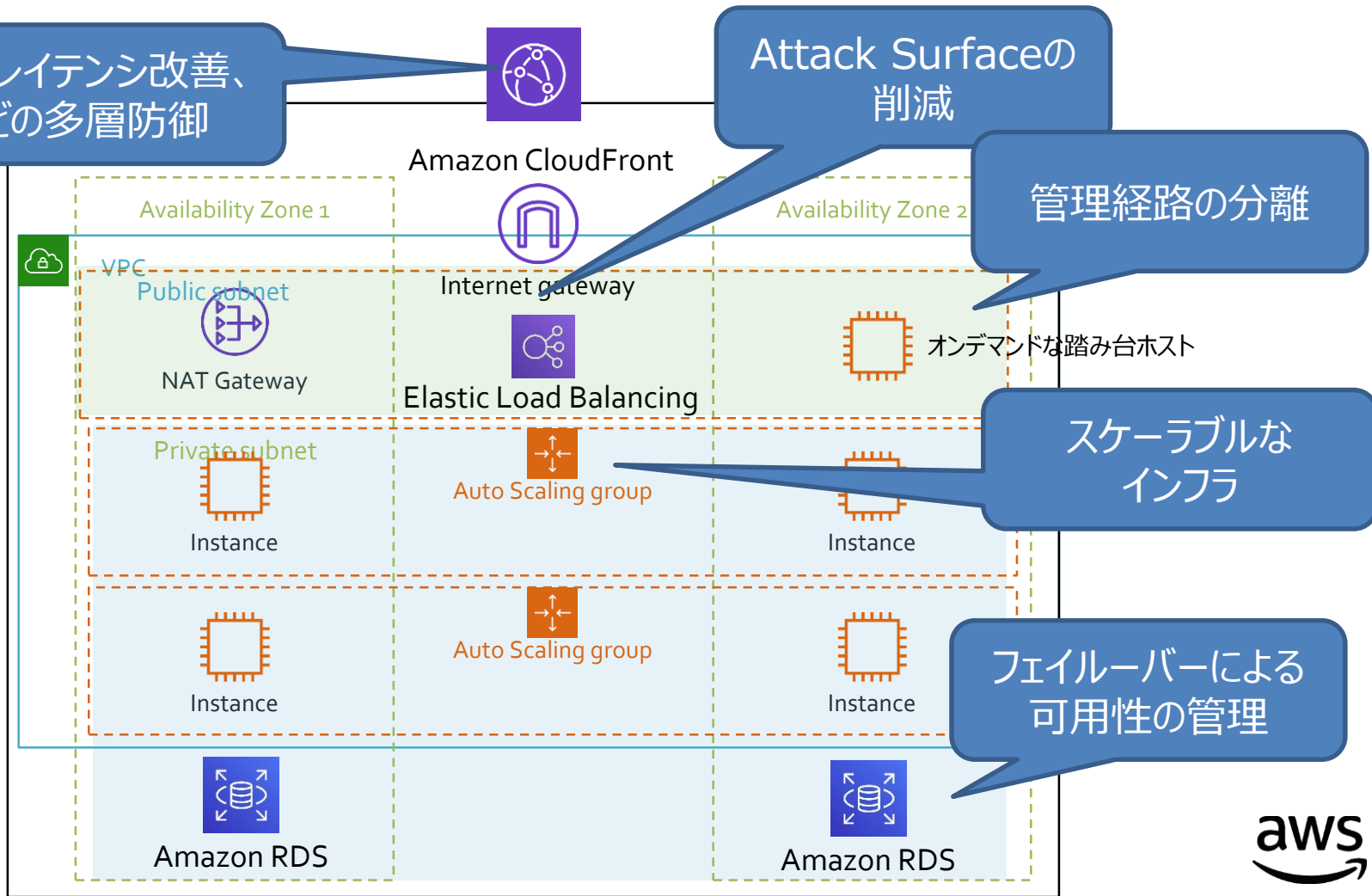
CDNによるレイテンシ改善、  
WAFなどの多層防御

Attack Surfaceの  
削減

管理経路の分離

スケーラブルな  
インフラ

フェイルオーバーによる  
可用性の管理



# AWS Systems Managerにみる管理の進歩



AWS Systems Manager

EC2インスタンスに対するコマンド発行（Run Command）を起点に、インベントリ情報の収集、SSH不要のサーバログイン、AMI（インスタンスイメージ）作成の自動化、パッチの適用などの様々な機能が追加



Automation



Documents



Patch  
manager



Parameter  
store



OpsCenter



Inventory



Maintenance  
windows



State  
manager



Run  
command



# アーキテクチャは何を目的にする？







# 適切なサービス選択による設計のメリット

- パッチ管理、可用性管理などをAWSにオフロード
- サービスの改善やデータセキュリティに注力
- 性質に応じたサービスを組み合わせる“Building block”の発想
- 内外のネットワークの意味？



# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ



AWS Cloud



## 応用問題

皆さんが実施したハンズオン、EC2上の静的なコンテンツはS3に配置することもできます。

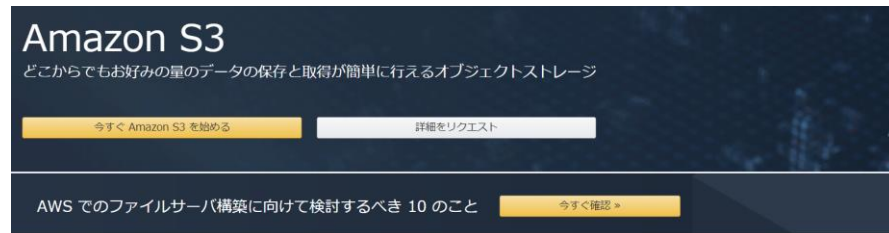
しかし、S3は3つのAZを前提として作られているサービスであり、VPC内には存在しません。

その場合、どうやってアクセスするのでしょうか。そのリスクは？



# ストレージであるAmazon S3を静的なウェブサイトに見せる

- デフォルトは非公開
- ストレージ容量は無制限
- 99.999999999999% (9 x 11) の耐久性を実現
- 99.9% の可用性を実現
- OS、ミドルウェア管理不要
- セキュリティ設定ミスに対する禁止機能、通知機能あり



Amazon Simple Storage Service (Amazon S3) は、業界をリードするスケーラビリティ、データ可用性、セキュリティ、およびパフォーマンスを提供するオブジェクトストレージサービスです。つまり、あらゆる規模や業界のお客様が、ウェブサイト、モバイルアプリケーション、バックアップおよび復元、アーカイブ、エンタープライズアプリケーション、IoT デバイス、ビッグデータ分析など、広範にわたるユースケースのデータを容量に関係なく、保存して保護することができます。Amazon S3 では使いやすい管理機能を使用するため、データを整理して、細かく調整されたアクセス制御を設定することで、特定のビジネスや組織、コンプライアンスの要件を満たすことができます。Amazon S3 は 99.9999999999% (9 x 11) の耐久性を実現するように設計されており、世界中の企業向けに何百万ものアプリケーションのデータを保存しています。



## 利点

業界をリードするパフォーマンス、スケーラビリティ、可用性、および耐久性

ストレージリソースをスケールアップ/ダウンして、変動する需要に対応します。先行投資やリソースの調達サイクルは不要です。Amazon S3 は、複数のシステムにまたがる S3 オブジェクトをすべて自動的に作成して保存しており、99.9999999999% (9 x 11) のデータ耐久性を実現するように設計されています。つまり、必要に応じてデータ

コスト効率の高いストレージクラス


パフォーマンスを下げずにコストを節約するには、対応するレートでさまざまなデータアクセスレベルをサポートする S3 ストレージクラスにデータを保存します。S3 ストレージクラス分析を使用して、アクセスパターンに基づき、低コストのストレージクラスに移動する必要があるデータを検出し、S3 ライフサイクルポリシーを設定して転送を実行できます。S3 Intelligent-Tiering では、アクセ

比類ないセキュリティ、コンプライアンス、監査機能

データを Amazon S3 に保存し、暗号化機能とアクセス管理ツールを使用して不正なアクセスからデータを保護します。S3 は S3 Block Public Access を使用して、バケットレベルまたはアカウントレベルで、すべてのオブジェクトへのパブリックアクセスをブロックできる唯一のオブジェクトストレージサービスです。S3 は PCI-DSS、

# VPCエンドポイントによるメリット、デメリット

- S3などをインターネットを経由せずに接続するサービス
- プライベートサブネットから直接対象に接続（他からの接続を制限することも）
- SaaSサービスをVPC内に直接接続させることも可能
- 一方、“使っていいサービス”の判定基準になることも。（使いたいサービスが使えない）



The screenshot shows the Amazon Virtual Private Cloud (VPC) User Guide. The left sidebar contains a navigation menu with the following items: Amazon VPC とは?, Amazon VPC の仕組み, 開始方法, VPC の例, VPC とサブネット, デフォルト VPC とデフォルトサブネット, IP アドレス指定, セキュリティ, and VPC のネットワーキングコンポーネント. The main content area is titled "VPC エンドポイント および VPC エンドポイントサービス (AWS PrivateLink)". Below the title are links for "PDF" and "RSS". The main text explains that VPC endpoints use AWS PrivateLink to connect to AWS services or VPC endpoint services privately, without needing an internet gateway, NAT device, VPN, or AWS Direct Connect. It states that VPC instances need public IP addresses for communication with other services, but VPC traffic remains on the Amazon network.

Amazon Virtual Private Cloud  
ユーザーガイド

Amazon VPC とは?  
Amazon VPC の仕組み  
▶ 開始方法  
▶ VPC の例  
▶ VPC とサブネット  
デフォルト VPC とデフォルトサブネット  
▶ IP アドレス指定  
▶ セキュリティ  
▶ VPC のネットワーキングコンポーネント

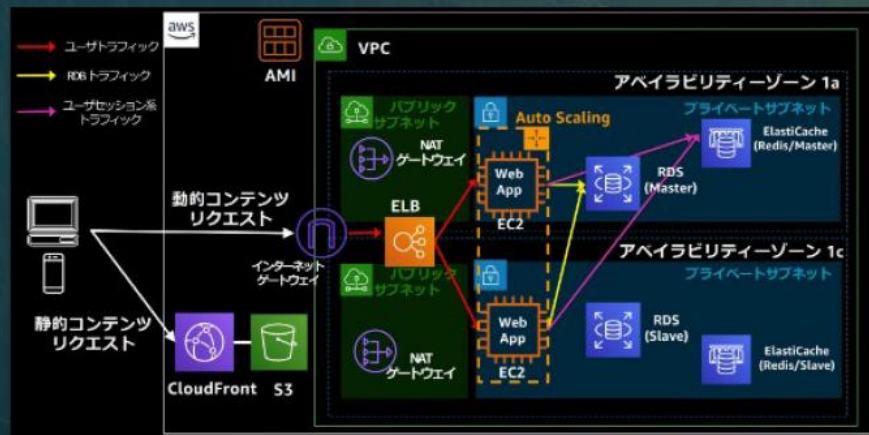
## VPC エンドポイント および VPC エンドポイントサービス (AWS PrivateLink)

[PDF](#) | [RSS](#)

VPC エンドポイントでは、AWS PrivateLink を使用する、サポートされている AWS サービスや VPC エンドポイントサービスに VPC をプライベートに接続できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、または AWS Direct Connect 接続は必要ありません。VPC のインスタンスは、サービスのリソースと通信するためにパブリック IP アドレスを必要としません。VPC と他のサービス間のトラフィックは、Amazon ネットワークを離れません。



# プライベートなネットワークの意味



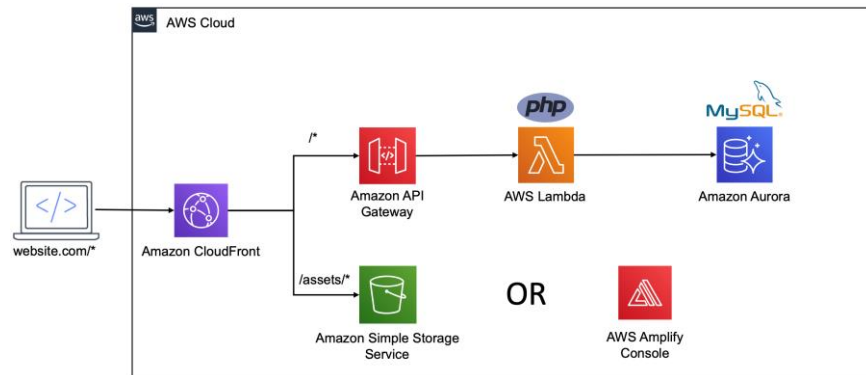
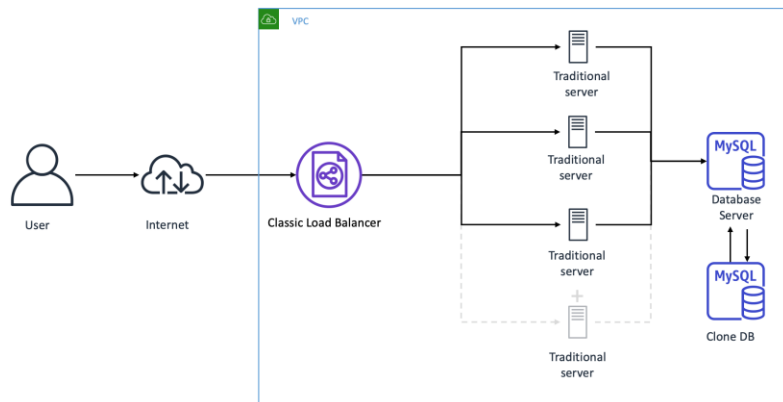
- S3やLambdaは基本的にはVPC外に存在
- アクセスコントロールはAWSのIAMで制御
- 経路は暗号化
- さて、リスクは大きいのか？





# サービスの実現方法の多様化

- 右記は、サーバレスアーキテクチャの組み合わせにより、運用者が実質的にサーバを管理しない構成
- サーバレスによりサーバやミドルウェアのアップデート等の管理負担からは解放
- どのような構成要素を選択するかは、組織の成熟度、サービスの目的、組織の意思決定による
- 現実的には、設計者のアーキテクティング能力への依存も



# ゼロ、イチではなくかわっていく“アクセスコントロール”

- ネットワークベースのアクセスコントロール：境界防御
- APIベースや属性ベースのアクセスコントロール：サービスに対するIdentity and Access management（この場合のIdentityはユーザだけではなくて様々なサービスやインスタンスであることも）

ゼロトラストのようなアーキテクチャの基礎になっている。

**ラボ 2**

# **Monitoring Security Groups with Amazon CloudWatch Events**



# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ

## ラボ 2

# Monitoring Security Groups with Amazon CloudWatch Events

- APIベースの世界：操作はログに。ログはイベントの契機に。

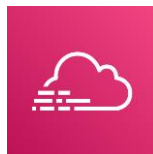
Security groupの操作→ Cloud Trail→ Cloudwatcevent

- サービスとしての監視基盤
  - たまり続ける大量のログ
  - 監視/通知をうけるための仕組みづくり
  - サーバとしてたてるなら、そのお守りが必要

- イベントドリブンのアーキテクチャ
  - 監視のための仕組みはイベント駆動
  - 人手を介さないオペレーション
  - サーバとしてたてるならそのお守りが必要



Event  
(event-based)



AWS CloudTrail



Lambda function



# 皆様に質問：サーバレスアーキテクチャを知っていますか





# サーバレスアーキテクチャがもたらす価値

- AWS Lambdaに代表されるアーキテクチャ
- 特定のコードをイベントに基づき実行
- 実行環境のメンテナンスが不要
- 各種サービスの連携や常時モニタリングのリソースが不要に



The banner features a dark blue background with a network of glowing blue lines. The text 'AWS Lambda' is prominently displayed in white. Below it, a line of Japanese text explains the serverless model. A yellow button labeled 'AWS Lambda の使用開始' is positioned below the text. At the bottom, an orange bar contains the 'aws SUMMIT ONLINE JAPAN' logo, the event title 'AWS Summit Online 2020 年 9 月に初開催決定!', a descriptive line about cloud technology, and another yellow button labeled '詳細はこちら'.

**AWS Lambda**  
サーバーについて検討することなくコードを実行できます。お支払いいただくのは、実際に使用したコンピューティング時間に対する料金のみです。

[AWS Lambda の使用開始](#)

**aws SUMMIT ONLINE JAPAN** **AWS Summit Online 2020 年 9 月に初開催決定!**  
クラウドの最新技術を "実際に手を動かして楽しみながら学ぶ" 無料クラウドカンファレンス [詳細はこちら](#)

AWS Lambda を使用することで、サーバーのプロビジョニングや管理することなく、コードを実行できます。料金は、コンピューティングに使用した時間に対してのみ発生します。

Lambda を使用すれば、実質どのようなタイプのアプリケーションやバックエンドサービスでも管理を必要とせずに実行できます。コードさえアップロードすれば、高可用性を実現しながらコードを実行およびスケールリングするために必要なことは、すべて Lambda により行われます。コードは、他の AWS サービスから自動的にトリガーするよう設定することも、ウェブやモバイルアプリケーションから直接呼び出すよう設定することもできます。



AWS Lambda とは (日本語字幕)

# AWSとは?

- Building block



# AWSとは? 様々な役割に応じたサービス



ビジネスアプリケーション



コンピューティング



カスタマーエンゲージメント



データベース



デスクトップとアプリケーションのストリーミング



開発者用ツール



Game Tech



IoT (モノのインターネット)



Machine Learning



マネジメントとガバナンス



メディアサービス



移行と転送



モバイル



ネットワーキングとコンテンツ配信



ロボット工学



人工衛星



セキュリティ、アイデンティティ、コンプライアンス



ストレージ



すべての製品を表示

# EC2 Auto Clean Room Forensics



# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ

# Amazon GuardDutyに見るサービスの進歩

## Amazon GuardDuty

インテリジェントな脅威検出と継続的なモニタリングで AWS のアカウント、ワークロード、データを保護

[Amazon GuardDuty の無料トライアルを開始する](#)

### 開発エンジニアのためのテクニカルカンファレンス

オードリー・タン氏、川口耕介氏、ジェームス・ゴスリンなど多数登壇！  
今おさえておくべきテクノロジーを網羅する 3 日間

aws DEV DAY  
JAPAN

[詳細・申込はこちら »](#)



Amazon GuardDuty は、AWS のアカウント、ワークロード、および Amazon S3 に保存されたデータを保護するために、悪意のあるアクティビティや不正な動作を継続的にモニタリングする脅威検出サービスです。クラウドでは、アカウントとネットワークのアクティビティの収集と集計こそシンプルになりますが、セキュリティチームが潜在的な脅威についてイベントログデータを継続的に分析するには多大な時間が必要となる場合があります。GuardDuty により、AWS での継続的な脅威検出のためにインテリジェントでコスト効率性に優れたオプションを活用できるようになります。このサービスは、機械学習、異常検出、および統合された脅威インテリジェンスを使用することで、潜在的な脅威を識別し、優先順位を付けます。GuardDuty は、AWS CloudTrail イベントログ、Amazon VPC フローログ、および DNS ログなどの複数の AWS データソース全体で何百億件ものイベントを分析します。GuardDuty は AWS マネジメントコンソールを数回クリックするだけで有効化でき、ソフトウェアやハードウェアをデプロイしたり維持したりする必要はありません。Amazon CloudWatch Events と統合することで、GuardDuty アラートは実用的となり、複数のアカウントにわたっての集計や、既存のイベント管理およびワークフローシステムへのプッシュを簡単にします。

Introduction to



Amazon GuardDuty

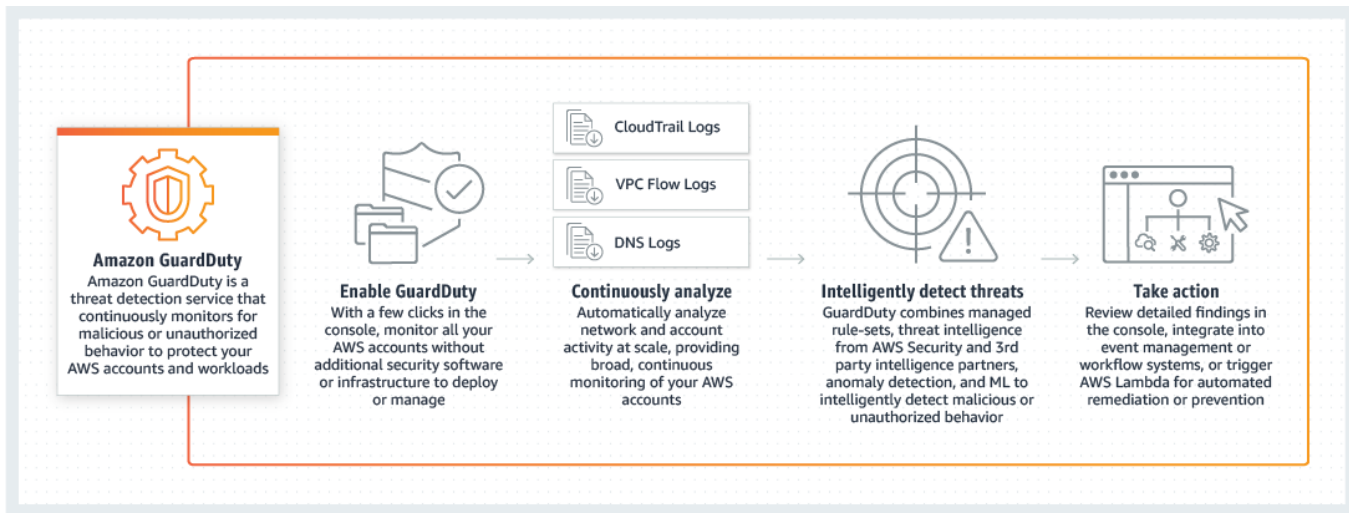
Introduction to Amazon GuardDuty (2:26)





# クラウドによるイノベーション (本来集中すべき作業は？)

## Amazon GuardDutyにみる変化 ログ収集→分析→脅威の通知→対応



# セキュリティサービスの進化

VPC flow logs

(ログ、モニタリング)

Guard Duty

(ログ分析のための機械学習を踏まえた脅威検知)

[Building block]  
セキュリティ自動化の実装  
(Lambdaによるイベント  
駆動型のリスク低減アクションの組み込み)

# モニタリングにみる規模の経済：

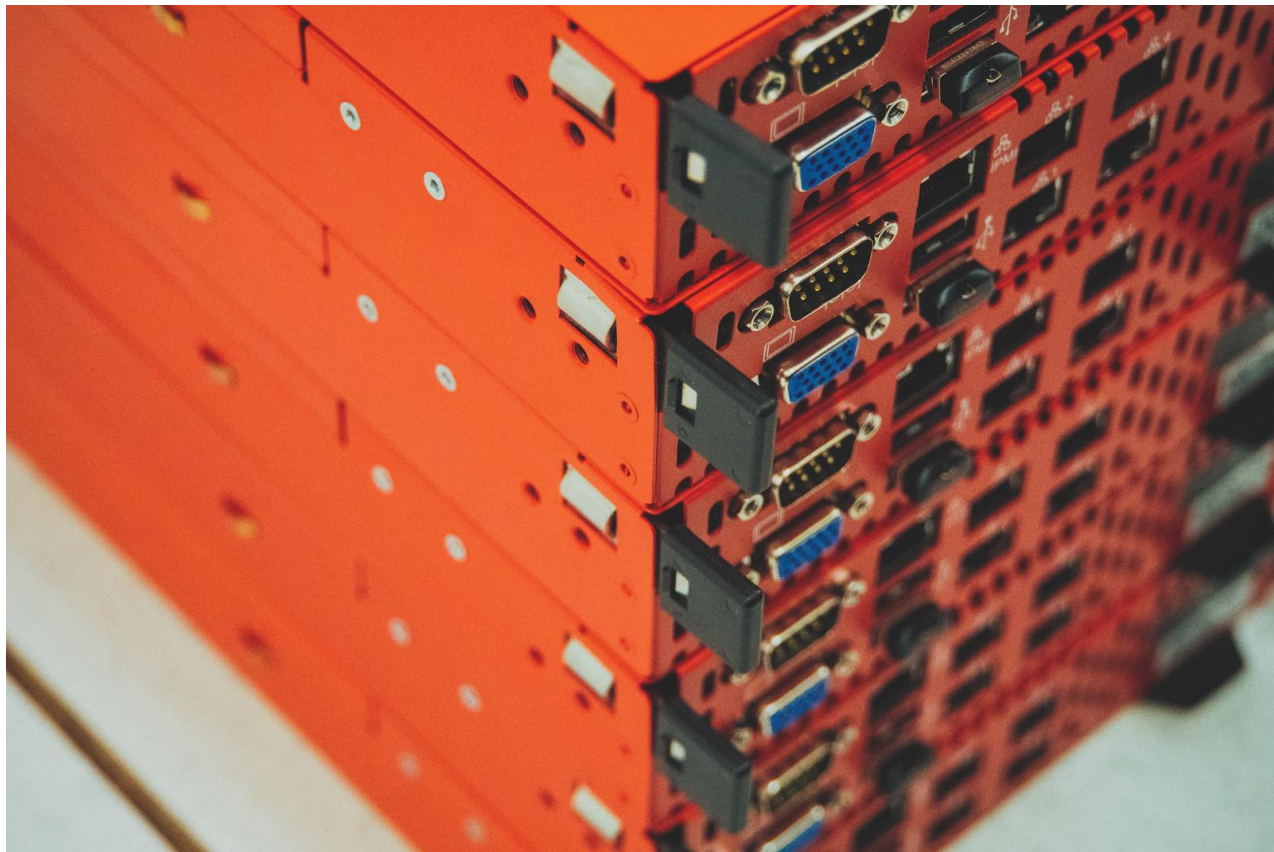
“ぼくがかんがえたさいきょうのきょういぶんせききばん”は来年も有効か

- 大規模な事業者に集まる大量の情報
- 大量の母集団に基づく分析
- サービス自体の継続的な成長
- “分析”“処理”のオフロード
- 組織がやるべき“意思決定”に集中



AIが人の仕事を奪う？ 機械ができる部分と人ができる部分を分業する？

# 監視基盤や脅威分析は“誰”が担うのか？



# AWSにおける発見的統制、状況の可視化

マネジメント  
コンソールでの  
目視やCLIなど  
(CIS Benchmark)



AWS Trusted Advisor



AWS Config



AWS Security Hub

AWSの提供する  
ベストプラクティスへの  
準拠

組織が選択できるマネージドルール、  
組織が開発できるルール

マルチアカウント統合や  
サードパーティ製品との  
統合ダッシュボード



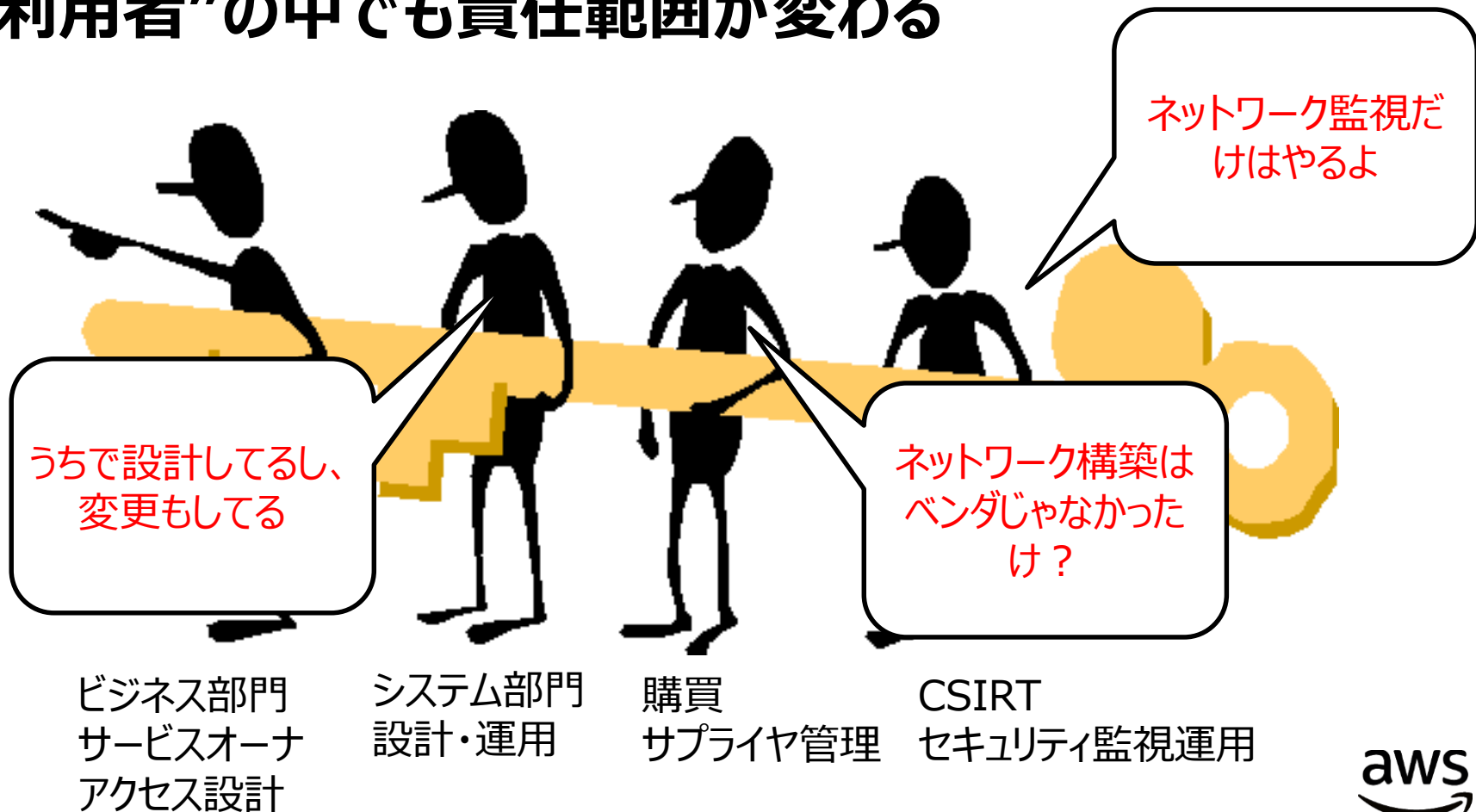
Auto remediation  
(自動修復)



Conformance pack  
(複数ルールのパッケージ化)



# “利用者”の中でも責任範囲が変わる



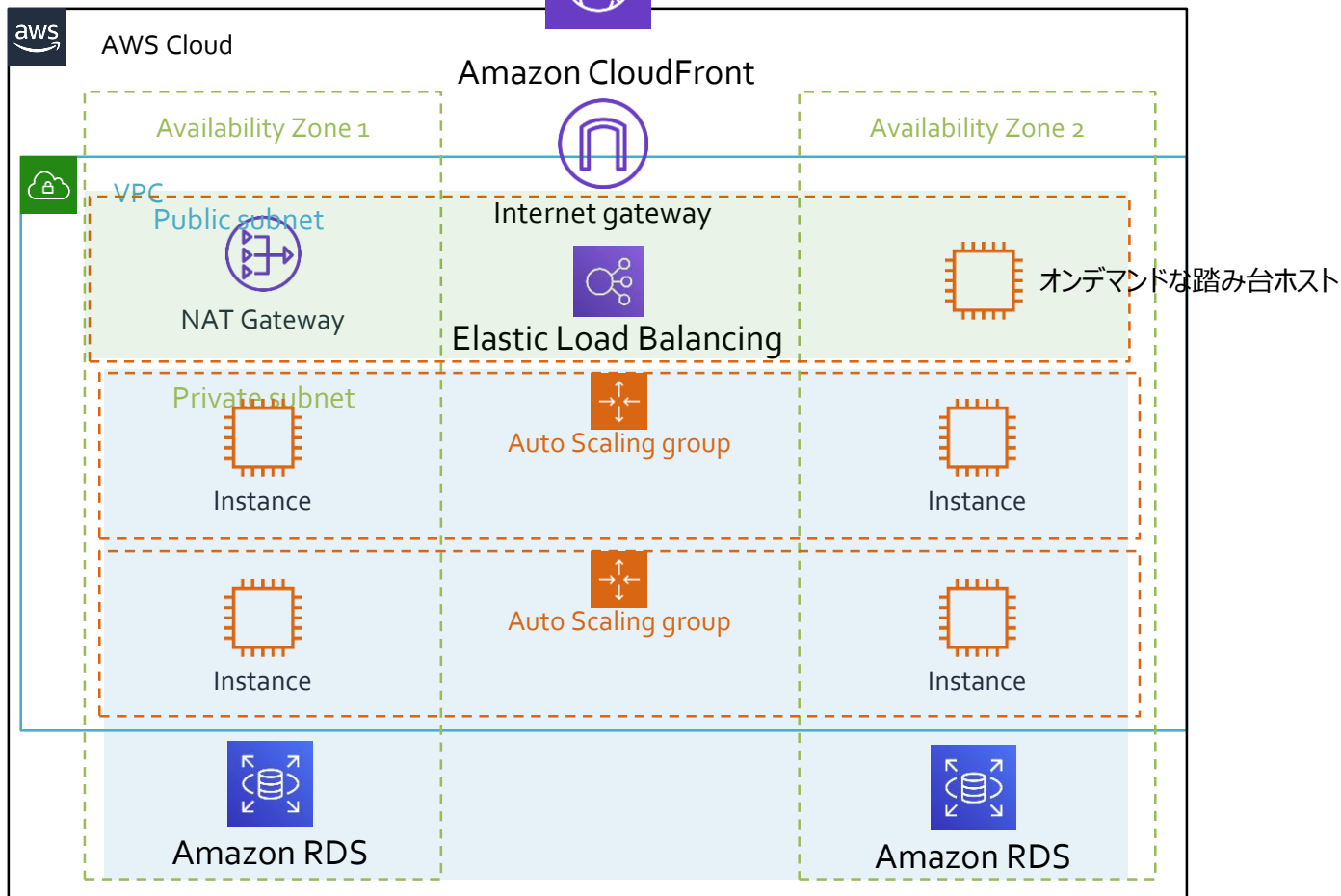
# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ





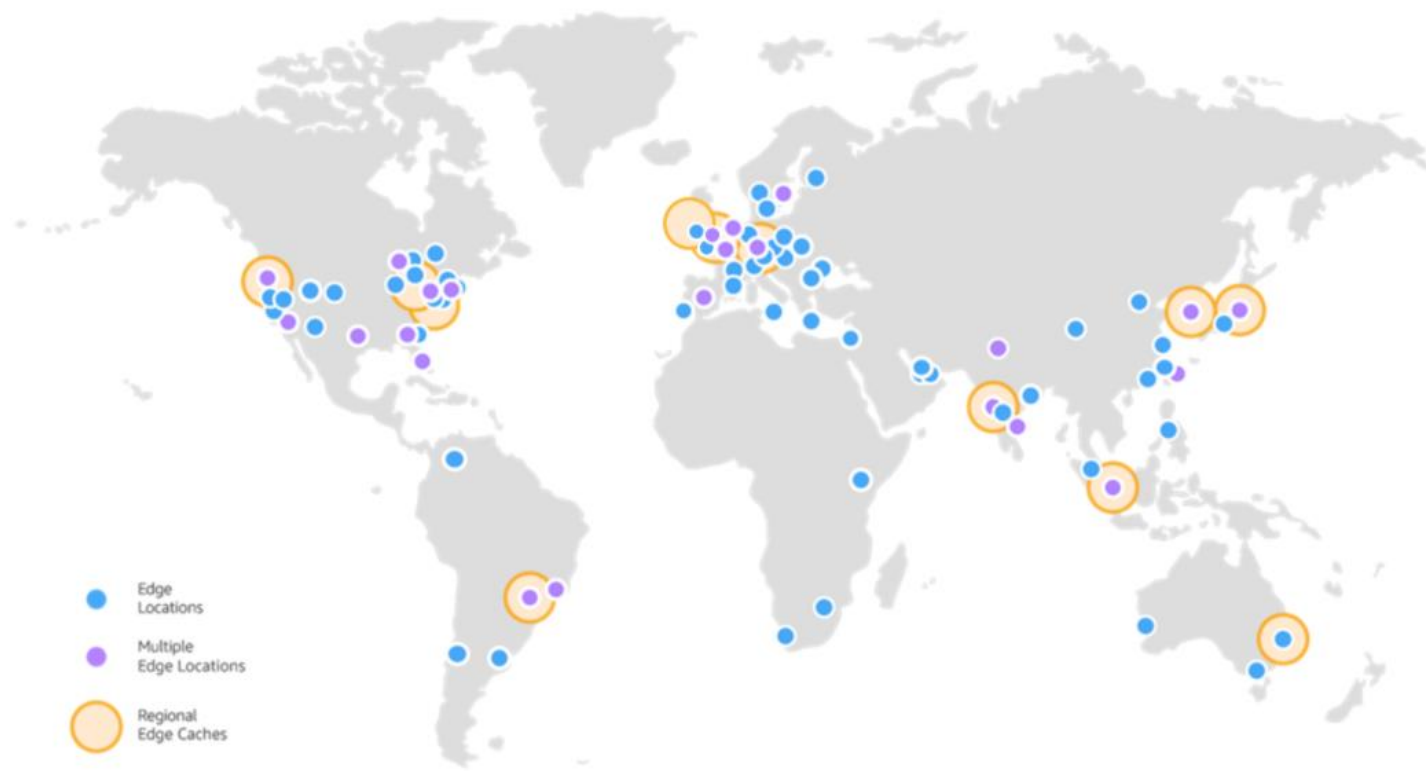
# Content delivery networkとWAFの実装



# Amazon CloudFront インフラストラクチャ

## Amazon CloudFront グローバルエッジネットワーク

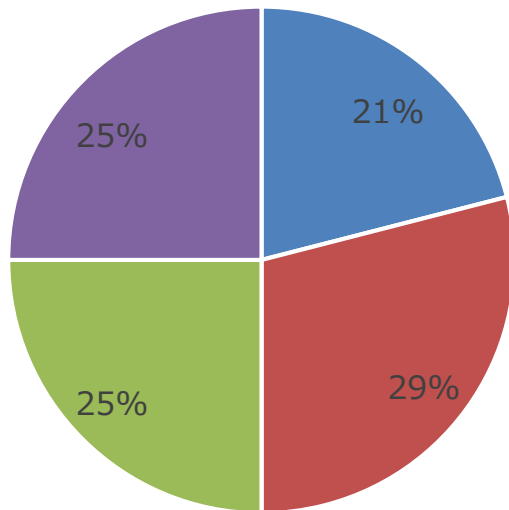
エンドユーザーにコンテンツをより低いレイテンシーで届けるため、Amazon CloudFront では 42 か国 84 都市にある 217 の POP (Point Of Presence) (205 のエッジロケーションと 12 のリージョン別エッジキャッシュ) のグローバルネットワークを使用しています。Amazon CloudFront エッジロケーションが設置されている国と都市は、以下のとおりです。



# AWSにおけるDDoS対策

## レイヤー別DDoS攻撃傾向

■ インフラレイヤ ■ ネットワークレイヤ ■ ネットワーク帯域レイヤ ■ アプリケーションレイヤ



### [1]インフラレイヤ

対象：DNSやCGNAT等

### [2]ネットワークレイヤ

対象：ファイアウォール、ルーターの飽和

### [3]ネットワーク帯域レイヤ

対象：回線帯域の飽和

### [4]アプリケーションレイヤ

サービスレイヤの飽和

お客様

クラウド内のセキュリティ  
に対する責任

お客様のデータ

プラットフォーム、アプリケーション、IDとアクセス管理

利用者の設計により防御

ネットワーク、ファイアウォール構成

クライアント側のデータ  
暗号化とデータ整合性  
認証

サーバー側の暗号化  
(ファイルシステムやデータ)

ネットワークトラフィック  
保護 (暗号化、整合性、アイ  
デンティティ)

AWS

クラウドのセキュリティに  
対する責任

ソフトウェア

コンピューート

ストレージ

データベース

ネットワーキング

ハードウェア/AWS グローバルインフラストラクチャー

AWSの責任範囲として防御

エッジロケーション

# AWS Shield の特徴

## ◀ ページの内容

### AWS Shield Standard

基盤となる AWS サービスの静的しきい値 DDoS 保護

インラインの攻撃緩和

### AWS Shield Advanced

アプリケーショントラフィックパターンに基づいてカスタマイズされた検出

正常性に基づく検出

高度な攻撃緩和機能

積極的なイベント応答

可視性と攻撃の通知

DDoS コスト保護

## AWS Shield Standard

### 基盤となる AWS サービスの静的しきい値 DDoS 保護

AWS Shield Standard では、AWS サービスへの受信トラフィックを検査し、トラフィックの署名、異常アルゴリズムおよび他の分析技術の組み合わせを使用してリアルタイムで悪意のあるトラフィックを検出する常時稼働のネットワークフローモニタリングを提供できます。Shield Standard は、AWS リソースタイプごとに静的しきい値を設定しますが、AWS のお客様のアプリケーションに対してカスタム保護を提供しません。

### インラインの攻撃緩和

自動化された緩和技術が AWS Shield Standard に組み込まれているため、頻繁に発生する一般的なインフラストラクチャ攻撃から AWS サービスを保護します。自動緩和策は、インラインで AWS サービスを保護するために適用されるため、レイテンシーの影響はありません。AWS Shield Standard では、決定論的なパケットフィルタリング、優先度を付けたトラフィックシェーピングなどの複数の技術を使用して、ベーシックなネットワークレイヤー攻撃を自動的に緩和します。

## AWS Shield Advanced

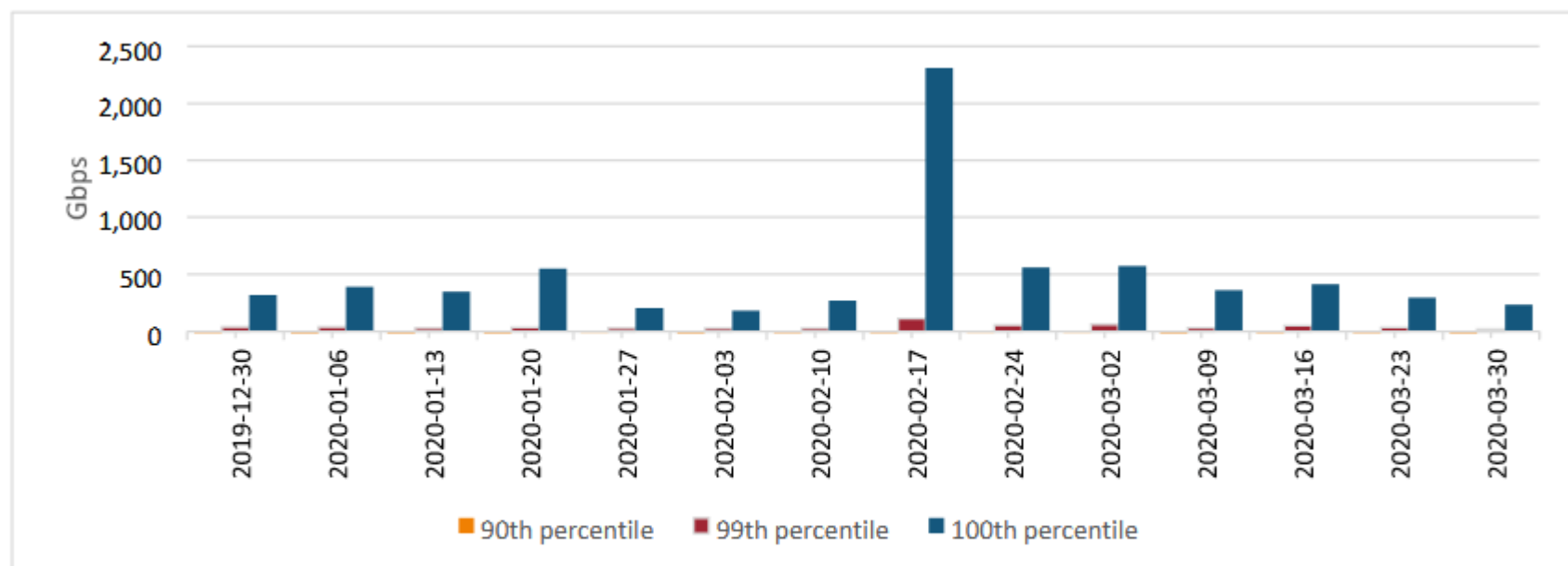


Figure 2. P90, P99, and P100 of volumetric events, measured in gigabits per second (Gbps), for resources on AWS during Q1 2020.

# 急激なトラフィックのスパイクはDDoSだけ？

[TOP](#) > WBS(ワールドビジネスサテライト)砲に備えよ！サイトの安定運用のためにやったことをまとめました



壽 かおり

2017年10月 4日 11:30

## WBS(ワールドビジネスサテライト)砲に備えよ！サイトの安定運用のためにやったことをまとめました

いいね! 95   シェア   ツイート   B! ブックマーク 9   Pocket 3

この記事のポイントは…

- ✓ テレビ番組出演に備え、放送中に一気に流入が来ると予想される自社サイトの安定稼働のためにCDN（コンテンツデリバリーネットワーク）を導入
- ✓ サッと導入してパッと戻せる、なるべく現構成に手を入れない一時的な負荷対策としてもCDNは有効
- ✓ 放映中とその後一時間ほど、普段より2桁多いトラフィックにも自社サイトを安定稼働できた



# アプリケーションレイヤへの攻撃

サービスの設計で受け止める

- トラフィックに応じてスケールアウト
- CloudFront (CDN) の活用

設計能力  
スケールアウトに対するコスト  
“Economic DDoS”

AWSのサービスで防御する

- CloudFrontによる接続元制限
- AWS Shield Advanced (AWS WAF)

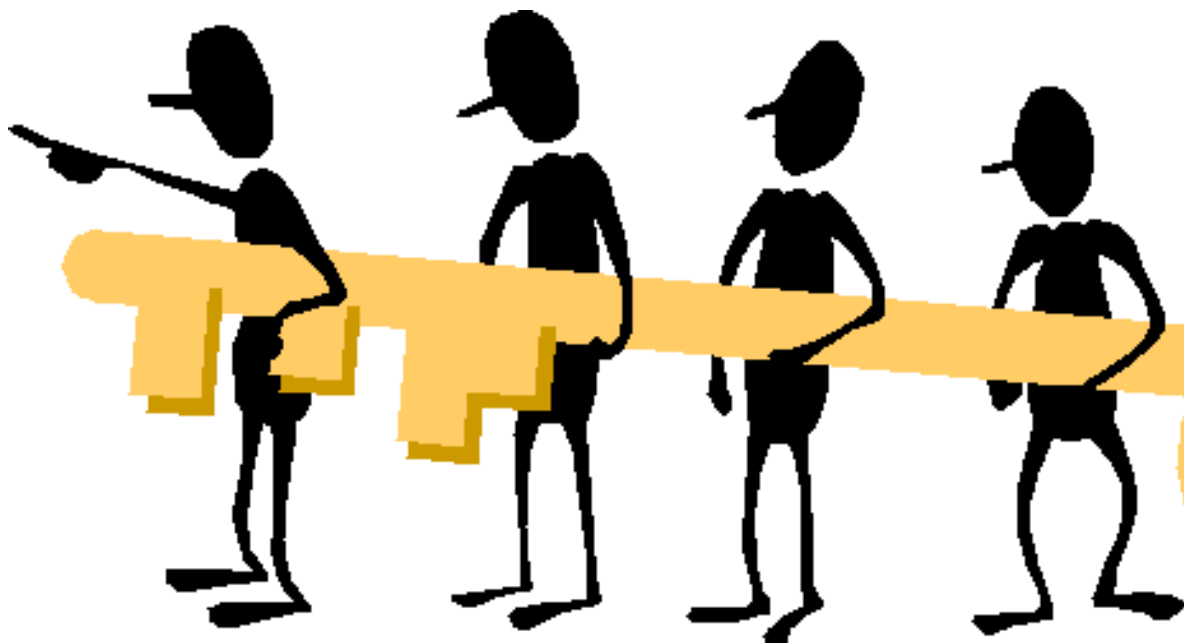
AWSサービスに対するコスト

様々なアプライアンスや  
パートナーソリューション  
の活用

- SaaS型のWAFサービスや他のCDN
- EC2上にサードパーティソリューションを設置

設計能力  
対応ベンダに対するコスト

# “利用者”の中でも責任範囲が変わる



どこまで自分たちで  
やるべきか

ビジネス部門  
サービスオーナー  
アクセス設計

システム部門  
設計・運用

購買  
サプライヤ管理

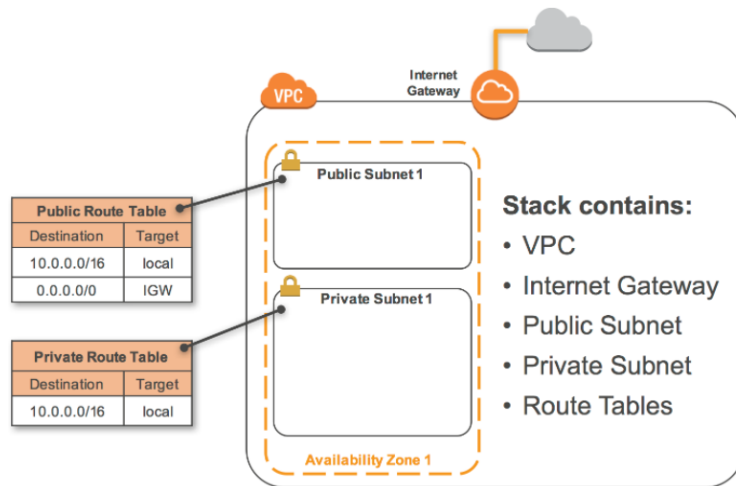
CSIRT  
セキュリティ監視運用

# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ

## ラボ 3

# Creating an Amazon Virtual Private Cloud (VPC) with AWS CloudFormation



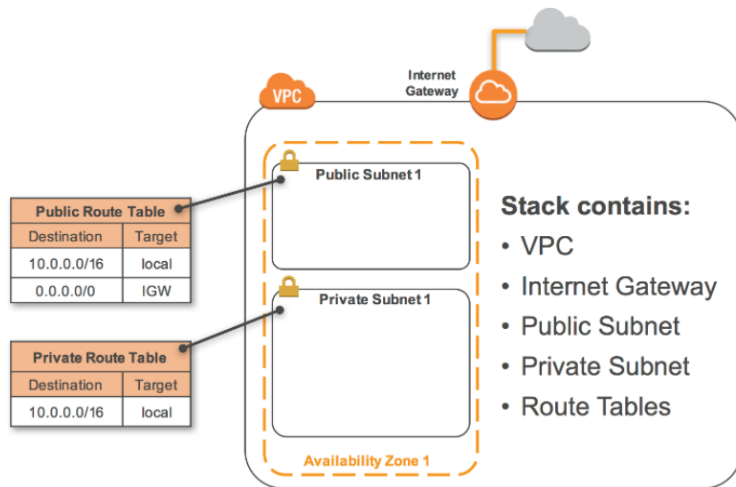
# ラボ 3

## Creating an Amazon Virtual Private Cloud (VPC) with AWS CloudFormation

- Infrastructure as codeの世界  
環境構築および変更の一貫性  
手作業による介入を削減

ラボ 1 に比べれば作業は簡単  
ただ、意味がわからないと価値もわからない。

コード化に対して“学習コスト”を払うに値するか



AWS CloudFormation



Template



Stack



# ゲートとガードレール



予防的な統制  
スピードをとめる  
ETCのような技術革新も



発見、訂正な統制  
スピードをとめず、危険時に効果

# Infrastructure as code

- 実装すべき環境はコードとして記述し、変更はコードに反映
- バージョン管理、テスト、障害時のリストア等はより容易に
- AWSのサービスとしてはCloud Formationが主に該当（単一のサービスではない）



## FC バルセロナの導入事例

2014 年

FC バルセロナ (FCBarcelona) は、スペインのバルセロナを拠点とする人気の高いサッカーチームです。開発チームは、SMS (ショートメッセージサービス)、e コマース、ソーシャルメディア、モバイルアプリを管理しています。このチームと共に働いている IT プロバイダーの Gnuine は、カスタムのウェブアプリケーションとモバイルアプリケーションを開発し、主にメディア方面でシステム管理サービスを提供しています。



AWS で、自前のプラットフォームの細部を管理する多大な手間とリソースを投入しなくて済むようになるため、枠組みが根本的に変わります」

Ramon Salvadó  
Gnuine 最高技術責任者

### 課題

Gnuine の主力製品の1つである Ubiquo Sports は、SaaS によるコンテンツ管理システム (CMS) で、FC バルセロナなどのスポーツ組織固有のニーズを解決することに焦点を絞っています。この Ubiquo Sports はすべてアマゾン ウェブ サービス (AWS) プラットフォーム上で動いているのです。

### アマゾン ウェブ サービスが選ばれた理由

FC バルセロナのオンラインマネージャーである Lluís Alsina 氏は、AWS の利用の決め手は伸縮自在性と従量制の価格モデルだったと言います。Gnuine の CTO の Ramon Salvadó 氏は、この決定に満足しました。「AWS に関しては、既にそれまでいくつかのプロジェクトでかなりの経験を積んでおり、そのすべてで優れた成果を上げていました」とコメントしています。さらに Salvadó 氏は次のように述べています。「当社だけでなくお客様にとっても、スケーリング、プロビジョニング、そしてセキュリティが重要です。AWS は容量に関して制約がないも同然で、それにも関わらず使用した分だけ料金を払えばよく、事前の投資が不要という点で、当然の選択と言えます。さらに、AWS はスポーツではよくあるトラフィックの急増にも、その柔軟性によって簡単に対応できます。」





# AWS利用成熟度の向上とガバナンスの強化



Amazon EC2

多くをマニュアルで  
統制、管理



AWS CloudFormation

ネットワークやAWS  
リソースのコード化



AWS Service Catalog

複数のテンプレートの  
管理効率化（セルフサービス化）

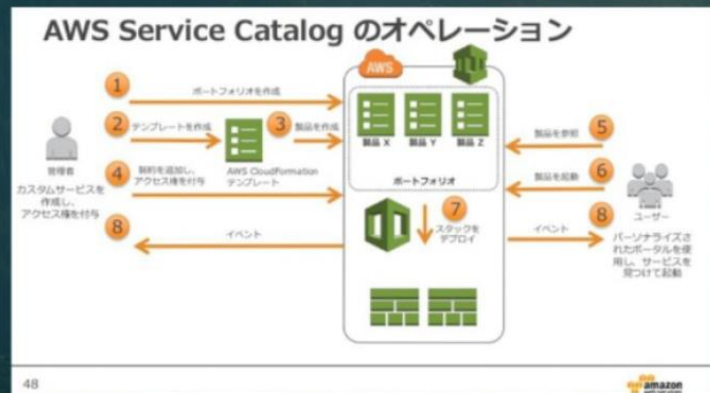


AWS Control Tower

マルチアカウント、  
アカウント管理の統合

# 中央集権・標準化はあるべき世界なのか？

Mentimeter



## ゆるやかな中央集権

- 利用者に一定の自由度を許容（サービスアジリティ）
- セキュリティをテンプレートに組み込み（Security By design）
- 継続的な監視、ガードレール化
- 環境の一貫性とモニタリングがあればリアルタイムに全件監査も現実的



**ガチガチに固めるよりも一定の自由度を許容する  
＝ゲート型からガードレール型のセキュリティへの変化**

# 100の環境があるとして監査戦略をどうやってたてますか？

グループ分けして監査対象を減らす

難しいですね。

環境をコード化して、差分を比較し一定の基準をもとに差分のデータを分析評価し、最適かつ効果的に監査できるサンプリング対象を抽出する

# 伝統的な“監査”やセキュリティの限界

- サンプルング
- 業務をとめない
- 乏しい人的資源



# 現在の情報セキュリティ監査の課題

- 拡張していくインフラ環境
- 複雑な構成要素
- 頻繁な環境の変更



# クラウドの監査に関するいくつかのアプローチ

- インベントリ情報の透明化
  - 動的な環境であり、紙面の設計書とのギャップがあるリスク
  - クラウド上の情報資産をリアルタイム把握することで正確性を担保
- テンプレート、イメージ化したインフラストラクチャ
  - コードの評価が監査結果となる。（評価の自動化による手作業からの解放）
  - 標準化されたテンプレートとその差分が評価対象となればサンプリングから解放
- コンプライアンスチェックの自動化（Compliance as code）
  - 評価すべき項目自体をサービスを通じて結果を取得し、時間効率、正確性を担保
  - 評価項目のカスタマイズが必要であれば自らでコード化、ツールを活用
  - 権限に対する注意（訂正的コントロールは監査人の範囲外）

→ 証跡収集を効率化し、そのワークロードをより有効な監査結論の分析へ





# CISBenchmarkにみる“コード化できるコンプライアンス”

- 様々なセキュリティのベストプラクティス項目に対し、Audit（監査方法）やRemediation（修正）をConsole操作やCLIでの記述方法を紹介
- 多くのサービスにおけるセキュリティ実装のベストプラクティスとして活用されている。

## 1.7 Eliminate use of the root user for administrative and daily tasks (Automated)

### Profile Applicability:

- Level 1

### Description:

With the creation of an AWS account, a *root user* is created that cannot be disabled or deleted. That user has unrestricted access to and control over all resources in the AWS account. It is highly recommended that the use of this account be avoided for everyday tasks.

### Rationale:

The *root user* has unrestricted access to and control over all account resources. Use of it is inconsistent with the principles of least privilege and separation of duties, and can lead to unnecessary harm due to error or account compromise.

### Audit:

#### From Console:

1. Login to the AWS Management Console at <https://console.aws.amazon.com/iam/>
2. In the left pane, click Credential Report
3. Click on Download Report
4. Open of Save the file locally
5. Locate the <root account> under the user column
6. Review password\_last\_used, access\_key\_1\_last\_used\_date, access\_key\_2\_last\_used\_date to determine when the *root user* was last used.

#### From Command Line:

Run the following CLI commands to provide a credential report for determining the last time the *root user* was used:

```
aws iam generate-credential-report

aws iam get-credential-report --query 'Content' --output text | base64 -d |
cut -d, -f1,5,11,16 | grep -B1 '<root account>'
```



# Introduction: customer story in re:invent session



A poster for an AWS re:INVENT session. The background is a dark, textured green with vertical streaks. A white rectangular border frames the central text. At the top, 'SID205' is written in white. Below it, 'AWS re:INVENT' is written in large, bold, white letters. Underneath, the title 'Building the Largest Repo for Serverless Compliance-as-Code' is written in white. Below the title, three speakers are listed: Gilles Ballet - Standard Chartered Bank - Head, Cloud and DevOps Architecture; Jonathan Rault - AWS - Security Lead APJAC, Professional Services; and Prashant Prahlad - AWS - Sr. Manager Product Management. At the bottom, the date 'November 30, 2017' is written. In the bottom left corner, the 'AWS re:Invent' logo is displayed. In the bottom right corner, the 'aws' logo is displayed.

SID205

## AWS re:INVENT

### Building the Largest Repo for Serverless Compliance-as-Code

Gilles Ballet - Standard Chartered Bank - Head, Cloud and DevOps Architecture  
Jonathan Rault - AWS - Security Lead APJAC, Professional Services  
Prashant Prahlad - AWS - Sr. Manager Product Management

November 30, 2017

**AWS re:Invent**  
© 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

[https://www.youtube.com/watch?v=VR\\_42ogewlo](https://www.youtube.com/watch?v=VR_42ogewlo)

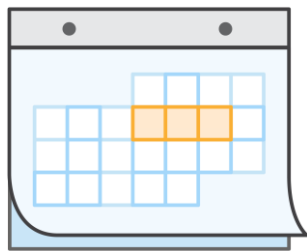




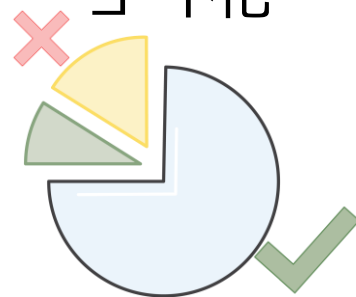
チェックリスト



チェックリスト自体の  
コード化



定期的（年  
1 ?）の監査



継続的な状態の  
可視化

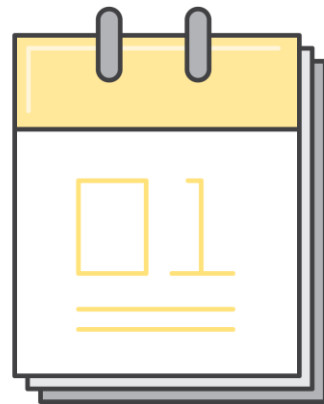
# 得られるメリット



価値のある時間と  
資源の活用



すべてのお客様への  
一貫性のある拡張



定期的な活動から  
毎日の活動へ

# セキュリティ現状の可視化



## AWS Security Hub

Quickly assess your high-priority security alerts and security posture across AWS accounts in one comprehensive view

**Amazon GuardDuty**

**Amazon Macie**

検出結果

アクション ▼    インサイトを作成する

検出結果は、セキュリティ上の問題か、または失敗したコンプライアンスチェックです。

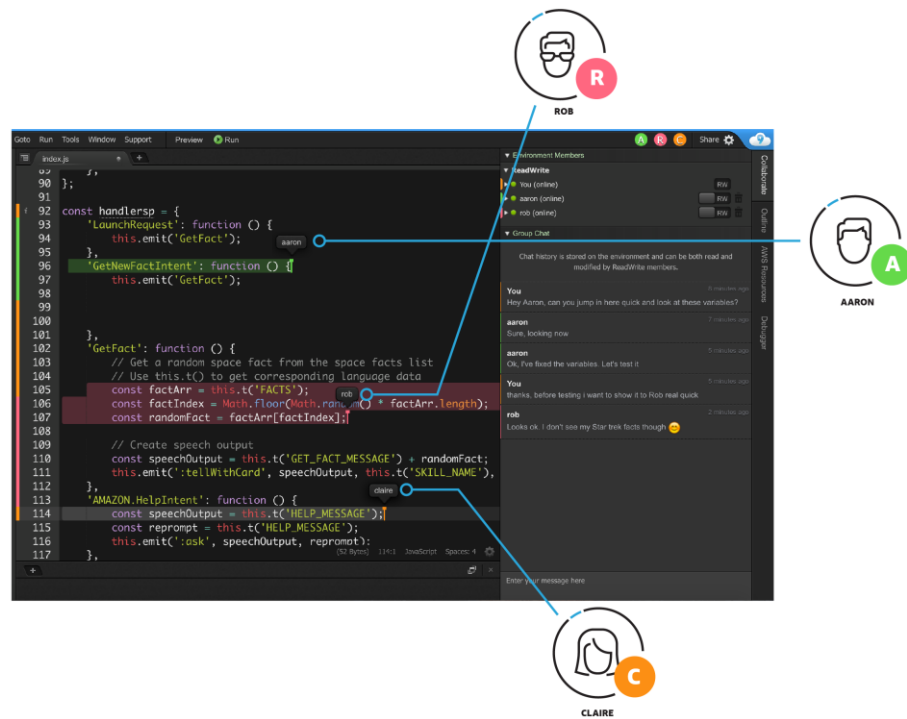
レコードの状態 EQUALS ACTIVE    フィルターを追加

< 1 ... >

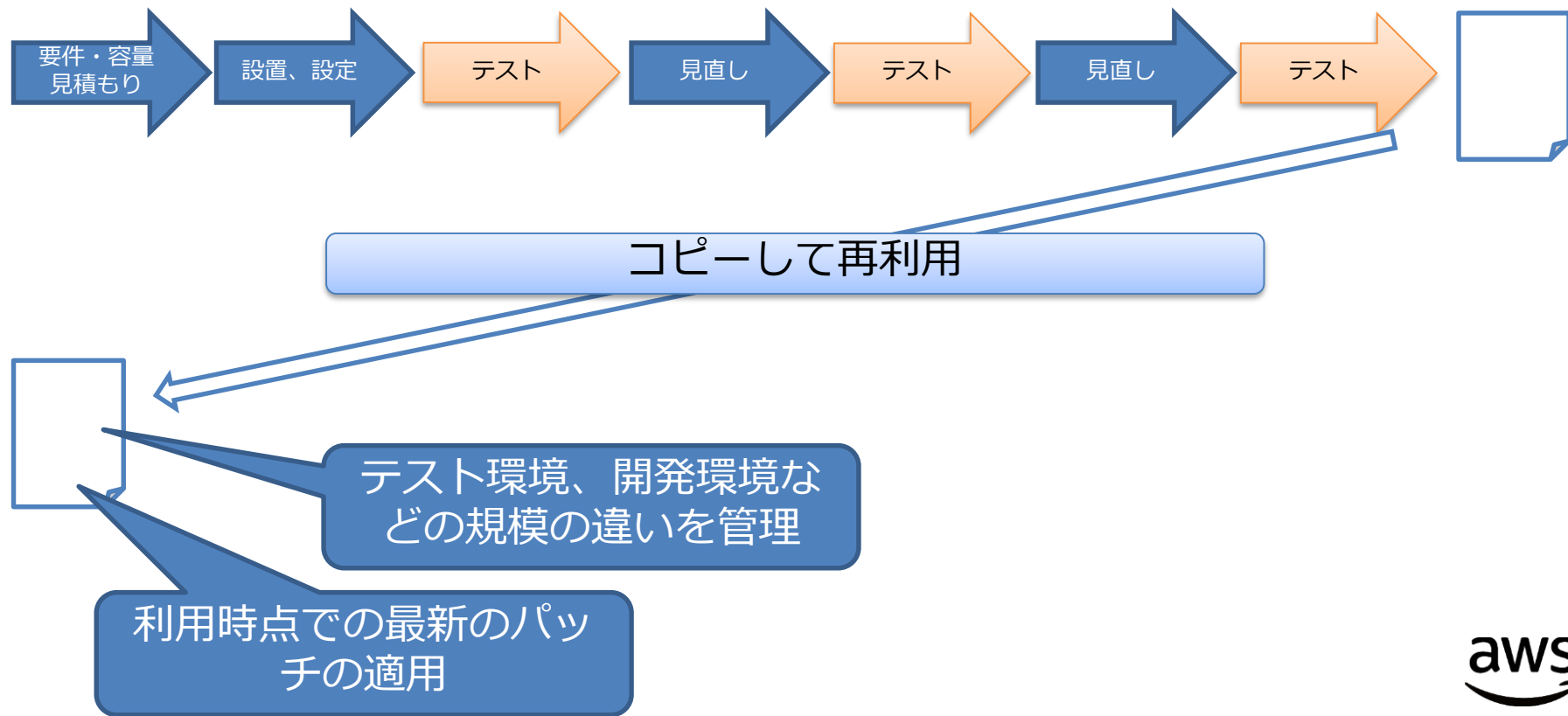
<input type="checkbox"/>	重要度 ▼	会社	製品	タイトル ▼	リソース ID	リソースタイプ	ステータス ▼	更新日時 ▼
<input type="checkbox"/>	HIGH	Personal	Default	[extra768] Find secrets in ECS task definitions variables (Not Scored) (Not part of CIS benchmark)	AWS::Account: 981647528212	AwsAccount	FAILED	14分前
<input type="checkbox"/>	HIGH	Personal	Default	[extra765] Check if ECR image scan on push is enabled (Not Scored) (Not part of CIS benchmark)	AWS::Account: 981647528212	AwsAccount	FAILED	14分前
<input type="checkbox"/>	HIGH	Personal	Default	[extra764] Check if S3 buckets have secure transport policy (Not Scored) (Not part of CIS benchmark)	AWS::Account: 981647528212	AwsAccount	FAILED	14分前
<input type="checkbox"/>	HIGH	Personal	Default	[extra764] Check if S3 buckets have secure transport policy (Not Scored) (Not part of CIS benchmark)	AWS::Account: 981647528212	AwsAccount	FAILED	14分前
<input type="checkbox"/>	HIGH	Personal	Default	[extra764] Check if S3 buckets have secure transport policy (Not Scored) (Not part of CIS benchmark)	AWS::Account: 981647528212	AwsAccount	FAILED	14分前

# 仮想化のもたらすもの：APIによる共通言語化

- すべてはプログラム
  - 文書化
  - 一貫性
  - 記録
  - 再利用
- APIの共通化
  - ネットワーク、サービンスタン  
ス等、ことなるアーキテク  
チャの一元化



# 仮想化のもたらすもの：実装までのリードタイムと品質





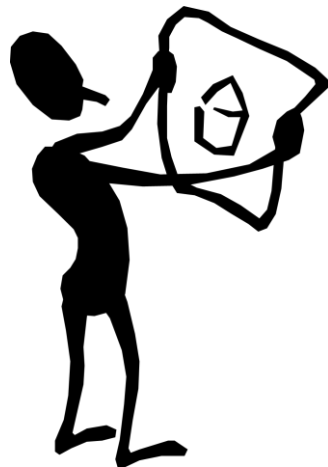
# コードは誰が書くの？

- プログラムコードとなると一般に学習コストが高まる
- 一方でGUIベースで手順書を作っても、ミスによるリスクの許容、UI変更への
- 継続的な対応が求められる

アプリケーションエンジニア

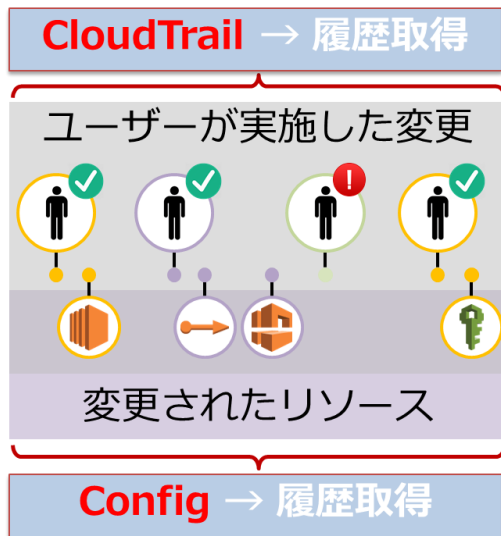
インフラエンジニア

フルスタックエンジニア



# 仮想化のもたらすもの：管理層のシフト

- 管理レイヤの分離：仮想化環境の管理、仮想化上の環境の管理者、権限が分離される（権限分離の明確化）
- 管理範囲の変化：従来の物理統制が利用者の統制として管理する範囲となる（**利用者が“ソフトウェア”として従来の物理管理を実装**）



## 管理範囲の変化例

- ネットワークの配線、構築
- お客様資産のサーバやネットワーク機器への物理的なアクセス
- インフラ管理者の作業記録

# DevOps: ITはニーズを踏まえて変わりゆくもの

- ハードウェア等の制約が強い環境ではITは変更弱い（そのためにウォーターフォールの設計、リリース後は最低限の変更）
- 本来ITは、利用者のニーズをふまえてサービスを提供するもの（柔軟性が必要）
- アジャイル的な手法をより容易に実現できるクラウドの登場により、フィードバックを踏まえたサービスのアップデートを行いやすい組織設計、ツールが台頭

## DevOps に AWS を使用する利点



**迅速に使用を開始する方法**  
AWS アカウントをお持ちのお客様は、すべての AWS のサービスをすぐに使用できます。事前のセットアップや、ソフトウェアのインストールは必要ありません。



**フルマネージドサービス**  
完全マネージドサービスを使用して、AWS リソースをより簡単に活用できます。インフラストラクチャのセットアップ、インストール、運用について心配する必要はありません。これによって主要な製品の開発に集中することが可能です。



**スケーリングを考慮した構築**  
AWS のサービスを使用して、単一のインスタンスを管理したり、数千単位までスケーリングしたりできます。これによって、プロビジョニング、設定、スケーリングを単純化し、ほとんどの柔軟なコンピューティングリソースの作成をサポートします。



**プログラム可能**  
AWS コマンドラインインターフェイス、または API や SDK 経由でそれぞれのサービスを使用できます。宣言型の AWS CloudFormation テンプレートを使用して、AWS リソースと、すべての AWS インフラストラクチャをモデリングしたり、プロビジョニングしたりすることも可能です。



**オートメーション**  
AWS では、オートメーションを使用して構築の高速化と効率化を達成できます。AWS のサービスを使用して、デプロイ、開発とテストのワークフロー、コンテナ管理、設定管理などの、手動で実行するタスクやプロセスを自動化できます。



**セキュア**  
AWS Identity and Access Management (IAM) を使用して、ユーザーのアクセス許可とポリシーを設定できます。これにより、リソースにアクセスできるユーザーや、リソースへのアクセス方法をきめ細やかにコントロールできます。



**大規模なパートナーエコシステム**  
AWS では、AWS のサービスに統合され、拡大する幅広いパートナーエコシステムをサポートしています。お好みのオープンソースのサードパーティツールを AWS と組み合わせ、エンドツーエンドのソリューションを構築できます。DevOps パートナーソリューションの詳細については、こちらを参照してください。



**使用した分だけ支払い**  
AWS では、使用を予定している期間のみ、必要とするサービスを購入します。AWS の料金には、初期費用、解約金はありません。AWS 無料利用枠を利用して AWS の使用を開始できます。詳細は各サービスの料金表ページを参照してください。



# Security by Design

- 求められるセキュリティ要件を設計として実装
- 開発、運用のプロセスにおいて反復的に評価
- 継続的でリアルタイムな監査の実現
- テンプレートの利用による一貫性の担保

## 設計によるセキュリティ

### 概要



AWS でのセキュリティ、コンプライアンス、およびガバナンスを自動化する

設計によるセキュリティ (SbD) は、AWS アカウントの設計の規格化、セキュリティ制御の自動化、および監査の合理化のためのセキュリティ保証アプローチです。SbD では、セキュリティを遡及的に監査するのではなく、AWS IT 管理プロセス全体にセキュリティ制御が組み込まれます。AWS CloudFormation で SbD テンプレートを使用することにより、クラウド内のセキュリティとコンプライアンスがより効率的かつ発展的なものになります。

SbD では、複数の業界、標準、およびセキュリティ基準にわたる大規模なセキュリティとコンプライアンスを実現するためのアプローチを採用しています。AWS SbD は、セキュリティのあらゆる段階でのセキュリティおよびコンプライアンスの機能を設計するために使用できます。設計の対象は、アクセス権限、ログ記録、信頼関係、暗号化の要求、承認されたマシンのイメージの要求といった AWS のお客様の環境内にあるすべてのものです。SbD により、お客様は AWS アカウントのフロントエンド構造を自動化し、AWS アカウントにセキュリティやコンプライアンスを信頼性の高い方法でコーディングし、IT 統制すべてにコンプライアンスを適用することができます。

### 設計によるセキュリティのアプローチ

SbD では、AWS で稼働する AWS のお客様のインフラストラクチャ、オペレーティングシステム、サービス、アプリケーションに関して、制御の責任、セキュリティベースラインのオートメーション、セキュリティの設定、およびお客様による制御の監査について概説しています。標準化および自動化され、規範的で再現可能なこの設計は、複数の業界やワークロードにわたる一般的なユースケース、セキュリティ基準、および監査要件でデプロイが可能です。

# Security by designのより容易な実現へ

- 様々なコードのテストツールやインシデント予防ツールが存在
  - Git-secret (認証情報の不要なアップロード防止)
  - Cfn-nag Cloudformationのセキュリティ評価ツール

このコンテンツは選択された言語でご利用いただけません。選択された言語でコンテンツをご利用いただけるよう現在 準備中です。ご不便をおかけしますが、しばらくお待ちください。

## Introducing AWS CloudFormation Guard (Preview) – a new open-source CLI for infrastructure compliance

Posted On: Jun 16, 2020

AWS CloudFormation announces the preview of AWS CloudFormation Guard (cfn-guard), an open-source command line interface (CLI) that helps enterprises keep their AWS infrastructure and application resources in compliance with their company policy guidelines. Cfn-guard provides compliance administrators with a simple, policy-as-code language to define rules that can check for both required and prohibited resource configurations. It enables developers to validate their CloudFormation templates against those rules.

Cfn-guard helps enterprises minimize risks related to overspending on operating costs, security vulnerabilities, legal issues, and more. For example, administrators can create rules to ensure that developers always create encrypted Amazon S3 buckets. Cfn-guard has a lightweight, declarative syntax that allows administrators to define rules quickly without needing to learn a programming language.

The administrators can also leverage a second open-source CLI called `cfn-guard-rulegen` to extract rules from existing compliant CloudFormation templates. With `cfn-guard-rulegen`, administrators don't have to create rules from scratch which speeds up the rules authoring process. The rules become a consistent record of compliant resource configurations that administrators can check into a source control such as GitHub to share across teams.

Developers can use cfn-guard either locally while editing templates or automatically as part of a CI/CD pipeline to stop deployment of non-compliant resources. If resources in the template fail the rules, cfn-guard provides developers information to help identify non-compliant resources.

AWS CloudFormation team welcomes feedback on the preview of AWS CloudFormation Guard and the contributions to the open source project. To get started, visit `cfn-guard` on GitHub.

[https://github.com/stelligent/cfn\\_nag](https://github.com/stelligent/cfn_nag)

<https://aws.amazon.com/jp/about-aws/whats-new/2020/06/introducing-aws-cloudformation-guard-preview/>



# Shift Leftやセキュリティバイデザインの実現へ

- ・問題があれば柔軟に変更できるサービス
- ・変更が新たな課題を生み出すことも

“後から評価するセキュリティ”から、  
“最初に設計し、常に評価するセキュリティ”

デザインレベルでセキュリティを設計  
設計を常にモニタリングする  
(デプロイパイプラインでの継続評価)  
インフラからアプリケーションまでを評価対象に

## AWS CloudFormation Guard (プレビュー) の紹介 – インフラストラクチャコンプライアンスのための新しいオープンソース CLI

投稿日: Jun 16, 2020

2020 年 10 月 1 日現在の更新: AWS CloudFormation Guard の一般提供を開始しました。

AWS CloudFormation は、AWS CloudFormation Guard (cfn-guard) のプレビューを発表します。これは、企業が AWS インフラストラクチャとアプリケーションリソースを会社のポリシーガイドラインに準拠させるのに役立つオープンソースのコマンドラインインターフェイス (CLI) です。Cfn-guard は、コンプライアンス管理者に、必要なリソース構成と禁止されたリソース構成の両方をチェックできるルールを定義するためのシンプルなコードとしてポリシー言語を提供します。これにより、開発者は CloudFormation テンプレートを各々のルールに対して検証できます。

Cfn-guard は、運用コスト、セキュリティの脆弱性、法的問題などへの過剰な支出に関連する企業のリスクを最小限に抑えることができます。たとえば、管理者はルールを作成して、開発者が常に暗号化された Amazon S3 バケットを作成するようにできます。Cfn-guard には軽量で宣言型の構文があり、管理者はプログラミング言語を習得することなくすばやくルールを定義できます。

管理者は、`cfn-guard-rulegen` と呼ばれる 2 番目のオープンソース CLI を利用して、準拠している既存の CloudFormation テンプレートからルールを抽出することもできます。`cfn-guard-rulegen` を使用すると、管理者は最初からルールを作成する必要がなくなり、ルールの作成プロセスがスピードアップします。ルールは、管理者が GitHub などのソース管理にチェックインしてチーム間で共有できる、準拠リソース構成に関する整合性のある記録になります。

開発者は、テンプレートの編集中にローカルで、または CI/CD パイプラインの一部として自動的に cfn-guard を使用して、非準拠リソースのデプロイメントを停止できます。テンプレート内のリソースがルールに違反した場合、cfn-guard は非準拠リソースの特定に役立つ情報を開発者に提供します。

AWS CloudFormation チームは、AWS CloudFormation Guard のプレビューとオープンソースプロジェクトへの貢献に関するお見をお待ちしております。はじめに、GitHub の `cfn-guard` にアクセスしてください。

# 一方で、新たに出る課題 = AWSのIaCツール？ ロックイン？

## Create reproducible infrastructure

- ✓ Reproducible production, staging, and development environments
- ✓ Shared modules for common infrastructure patterns
- ✓ Combine multiple providers consistently

Terraform makes it easy to re-use configurations for similar infrastructure, helping you avoid mistakes and save time.

[View All Providers](#)



<https://www.terraform.io/>





# AGENDA

- ラボ1 振り返りより
  - クラウドセキュリティの考え方
  - ゼロイチではなくかわっていくアクセスコントロール
- ラボ2 振り返りより
  - サーバレスアーキテクチャがもたらす価値
  - Amazon GuardDutyに見るサービスの進歩
  - DDoSと設計：利用者の責任範囲を考える
- ラボ3 振り返りより
  - Infrastructure as codeとCompliance as code（ガードレール型のセキュリティへ）
  - ロックイン、というリスクへのアプローチ

**ロックインという“リスク”へのアプローチ**

DEV DAY  
TOKYO

F-3

# 開発におけるロックインの リスク評価と考え方

Fumihiko Hata  
Solutions Architect  
Amazon Web Services Japan



2019.10.03-04

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

<https://pages.awscloud.com/rs/112-TZM-766/images/F-3.pdf>

# ベンダーロックイン

- あるベンダーの製品 A を採用した。時間が経ち、その製品 A をやめて他社の別製品 B に変更したくなった。しかしそれができない、あるいは多大な労力を要する。

# What is lock-in?

- ベンダーロックイン：

- ロックインと言うときはこれを指すことが多い。前ページで説明したような状況。

- プロダクトロックイン：

- ベンダーロックインの状況において大抵はプロダクトロックインにもなっている。
- ただし、ベンダーがないプロダクト、例えば **OSS** でもプロダクト・ロックインは起きうる。**Cassandra** を使い倒して何年も経ってデータ量も多い状態から **HBase** に移行しようとする、移行はどれくらいの労力でしょうか。この場合、ベンダーはいません。しかし、ベンダーロックインと同様に開発コミュニティにロックインしているという捉え方も。

- プラットフォームロックイン：

- 単一の製品ではなく、プラットフォームの製品群やそのサービス全体に対するロックイン

- アーキテクチャロックイン：

- サーバーレスアーキテクチャをコンテナベースのアーキテクチャに変えるとしたら？
- マイクロサービスとモノリスではどちらがアーキテクチャ変更しやすいでしょうか？

Gregor Hohpe の『Don't get locked up into avoiding lock-in』の記事では、これ以外に **Skills Lock-in**, **Mental Lock-in**, **Version Lock-in** など挙げより詳細な考察をしています。

<https://martinfowler.com/articles/oss-lockin.html>

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



# DEV DAY

“「ロックイン」という用語は誤解を招きます。私たちはただ switching costs の話をしています。switching costs は IT の歴史を通じて常に存在していました。プラットフォームまたはベンダーにコミットすると、その瞬間に、後から変更する場合の switching costs が生まれます。Java を選択してから Node.js に移行すればコストがかかります。（中略）そこには単に switching costs があるだけです。状況によってそのコストは大きくも小さくもなります。”

Mark Schwartz, Enterprise Strategist AWS  
『Switching Costs and Lock-In』

元 US Citizenship and Immigration Service の CIO

『a Seat at the Table』, 『THE ART OF BUSINESS VALUE』 著者





# 開発にまつわる様々な変更コストを下げる

1. 技術的負債と向き合う  
→ コード、インフラ、アーキテクチャ、そして組織の変更コスト下げる
2. CI/CD パイプライン  
→ ソフトウェア（コード）の変更コストを下げる
3. Infrastructure as Code  
→ インフラストラクチャの変更コストを下げる
4. 疎結合なアーキテクチャ  
→ アーキテクチャの変更コストを下げる

※ 実際はインフラ構成を CI したりもする。上記は分かりやすく分類している。



可能性が低い乗り換えにおける switching cost を  
過剰に見積もり過ぎないでください

実際の現実的な可能性や確率でかけ算してあげてください

それによって失われるもののリスクも評価してください  
(言い換えれば、得られる利益)

# ロックインにみる人材育成のチャレンジ

変化の激しいクラウドの利活用には、サービスの中でもロックインが存在し、管理する必要がある

- ロックインもセキュリティマネジメントも“有無”ではなく“高低”による定量化を考える（リスクベースアプローチ）
- 設計次第でコストは変動する（Controllable）
- 現状のアーキテクチャが常に正しいとは限らない（ベストプラクティスの陳腐化、新機能による実現範囲の変化）
- ロックインを避けるコストと学習コストのバランス



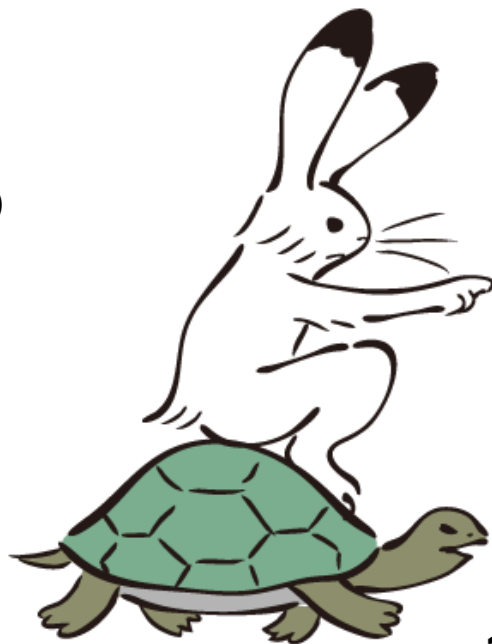
<https://aws.amazon.com/jp/blogs/news/webinar-bb-anti-pattern-2018/>



# ウサギと亀

クラウドが求められているのは“サービスが変化”しなければいけないから

- スタートアップ
  - ロックイン上等でまずは使って、試す。
    - 知識の高まりとともにサービスを変化させる
    - 自社がどうしたか、を伝える（事例化による組織価値向上）
    - 組織的な経験値や知見の欠落（個人スキルへの依存）
- エンタープライズ
  - 初期投資に非常に慎重
    - 他社はどうしているか？（事例、先例）
    - 評価、ポリシーにおける組織的知見の文書化
    - 技術的負債の束縛



# ウサギ的な亀になれるのか？

組織は学習コストを適切に管理しているか

組織は不確実性の高い学びのスポンサーになれるか

個人は組織にどこまで、何を期待するか

コミュニティを通した学びの価値と発展は？



**おわりに：何を伝えたかったかをあらためて**

# ブレイクアウト：講義での気づきシェア

まずは個人でひとりブレスト（書き出してみる）

ー＞ つぎにグループで意見を出し合う

ー＞ 出た意見を自分でまとめてみる。後でアンケートにかけるように

# 技術におけるクラウドとセキュリティ

- クラウド的なアーキテクチャを生かしたセキュリティ
  - 様々な選択肢
  - 疎結合
  - スケーラブルなインフラストラクチャ
  - Infrastructure as code
  - 進化するサービスと規模の経済
- ガードレール型のセキュリティ
  - 予防的な統制（ゲート）に発見・定性的な統制の組み込み
  - クラウドがもたらした発見・定性的な統制の民主化

# DevSecOpsは“文化”、これってなんだろう

## 文化＝組織としての学びを仕組み化するための土台

- ウサギとカメの対立
  - 新たな技術によりサービスや運用を改善したいタイプのエンジニア
  - リスクが高いことに対しては慎重になるセキュリティやリスク管理部門
  - 学習に対する姿勢やモチベーションの違い
- カメはサボらないウサギに勝てない現実
  - 組織の許可する範囲でのみ学ぶタイプのエンジニア
  - リスク分散を踏まえて広く浅く教育機会を提供するマネジメント
- 温度がわかるカエルの必要性
  - 違うチームをつなぐ“疎結合”な人材の必要性（橋渡し人材？）





ご清聴ありがとうございました。  
質問、感想タイムへ。



**aws Security Roadshow Japan**

[今すぐ申し込む >>](#)

### イベント概要

「AWS Security Roadshow」は、クラウドのセキュリティ・コンプライアンスに関する最新の情報をお客様にお届けすることを目的としています。

現在、多くの企業や組織ではテレワーク、リモートワークに移行しており、「ゼロトラスト」に対する議論の活発化等、セキュリティモデルも大きな変革の時期を迎えています。

本イベントでは、お客様が抱える様々なセキュリティに対する課題とその対応方法、実際のお客様のAWSのセキュリティサービス活用事例、今後必要となるセキュリティ対策、AWSのセキュリティの方向性などを包括的にご説明する予定です。

企業・組織の中で、セキュリティやコンプライアンスに課題をお持ちの皆様（営業、マーケティング、開発や運用など）、セキュリティに関する意思決定に関わる管理職や役員の方、公共機関や金融機関のセキュリティ担当者など、AWSセキュリティの最新情報を知りたい方は是非ご参加ください。

日時： 2020 年 10 月 28 日(水) 10:00-16:30  
会場： オンライン開催  
参加費： 無料（要事前申込）  
定員： 500名

※お申し込みが多数の場合には、定員になり次第締め切らせていただきます。

