

公共機関における AWS 公開テンプレート の活用

～ Infrastructure as Code の概要と活用 ～

梶木 正博

パブリックセクター技術統括本部 ソリューション アーキテクト
アマゾン ウェブ サービス ジャパン合同会社

自己紹介

榎木 正博 (たぶき まさひろ)

- ・ 公共に属するお客様を担当している
パブリックセクター 技術本部にて主に自治体の
お客様のクラウド活用支援を担当
- ・ 前職は仮想化ソフトウェアベンダーに在籍、
自治体・大学・病院のお客様のほか、通信キャリアの
お客様の技術支援を担当
- ・ オンプレミスの仮想化の知識もあわせて
自治体のお客様のクラウド移行支援や
新しいクラウドの活用のお手伝いをしています

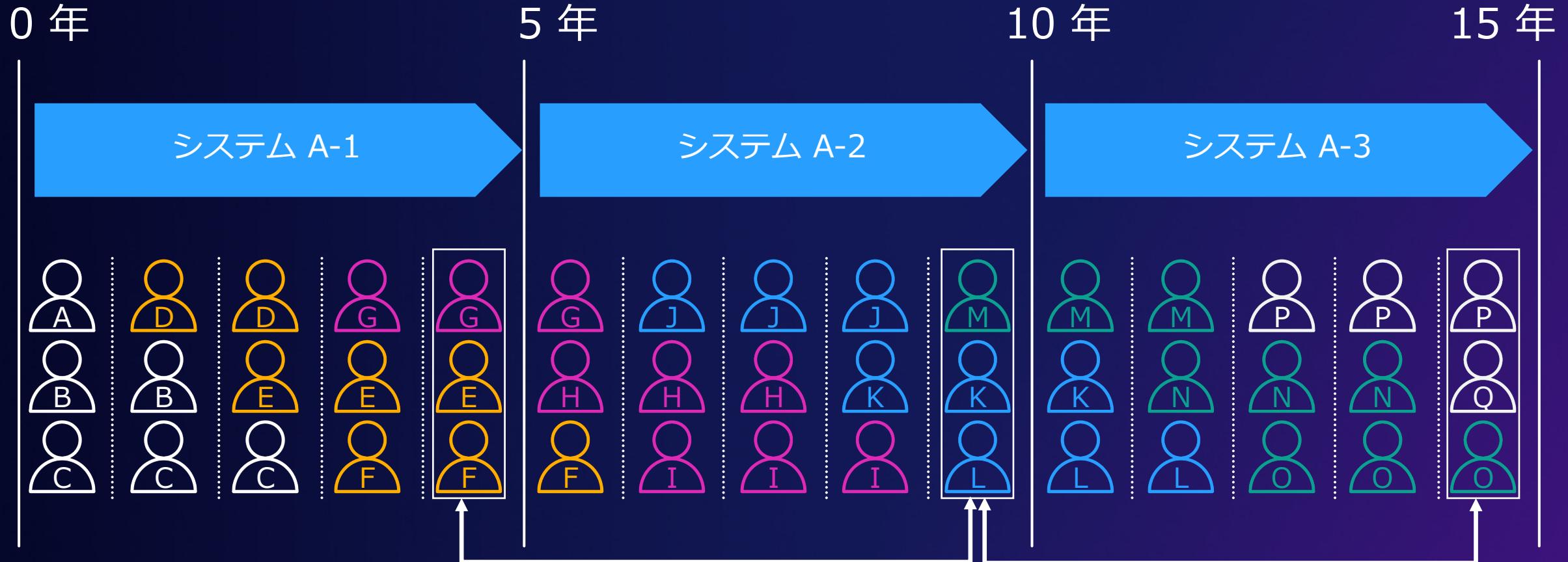


Agenda

- 公共機関におけるクラウド利用の課題
- クラウド上での統制 – IaC を使ってクラウドをあるべき状態に保つ –
- テンプレートを使った統制 – Baseline Environment on AWS –

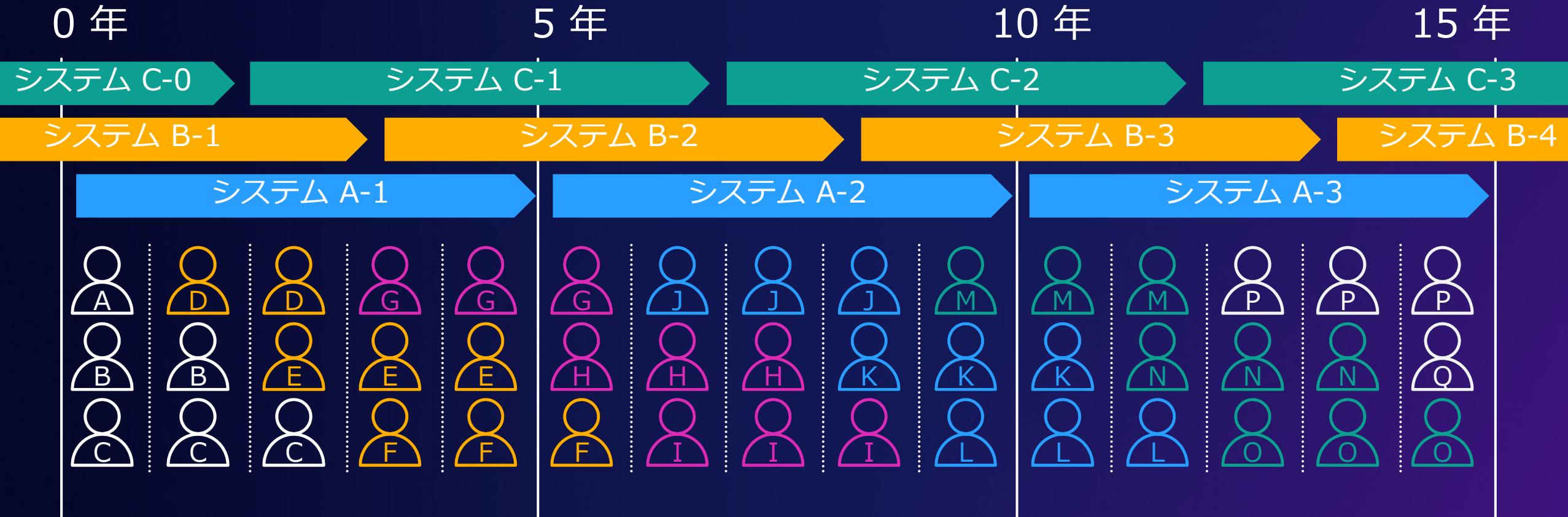
公共機関におけるクラウド利用の課題

システム更改時に既存システムの検討事項が曖昧に



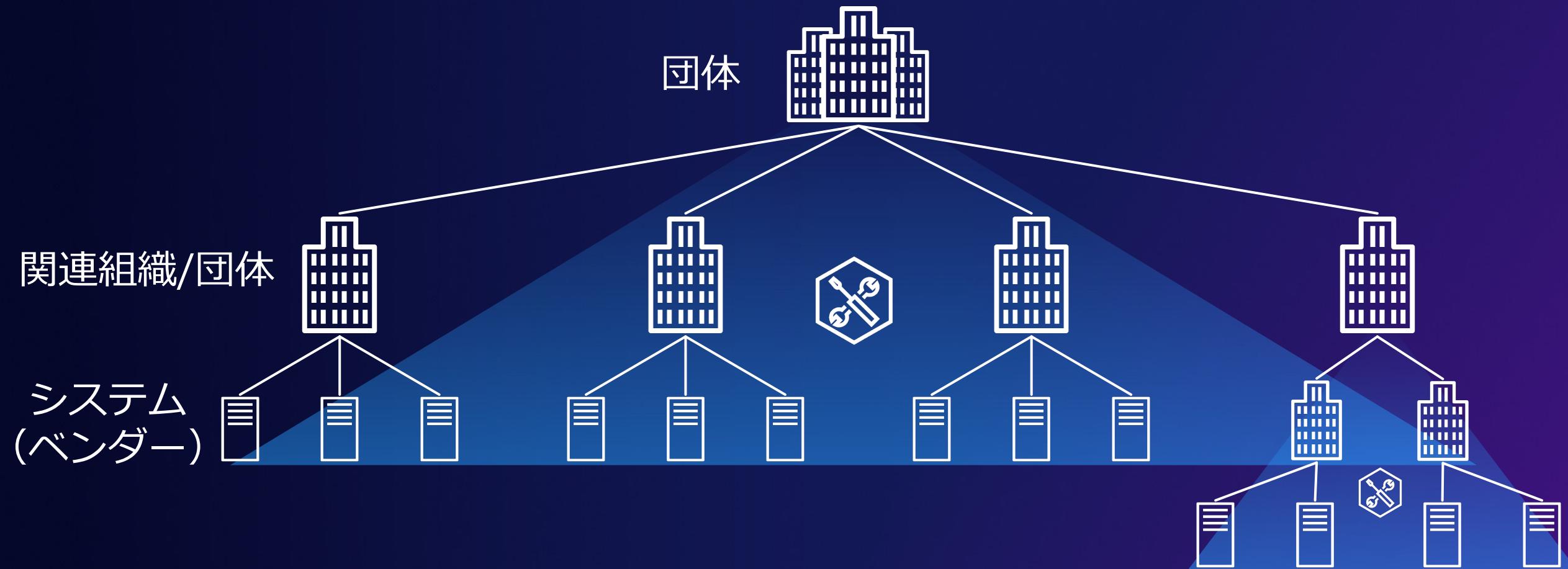
部署異動や退職などによって、システム更改時において
既存システムで検討していたポイントが把握しにくい

システム更改時に既存システムの検討事項が曖昧に



一つの部署で複数のシステムを管理することも多く、
さらに各システムの情報を把握・共有・伝達することが困難に

マルチベンダー・マルチテナント環境での統制



複数団体や複数ベンダーを抱える組織において
システムを構築する際のポリシーを共通化しておくことが重要

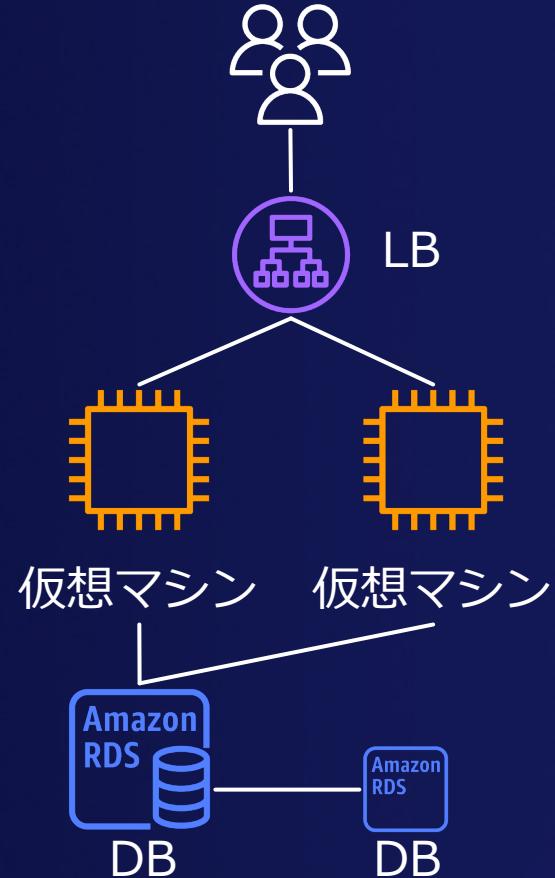
クラウドでシステムを構築する場合の選択肢は様々



仮想マシン

可用性：低

マネージド使用率：低

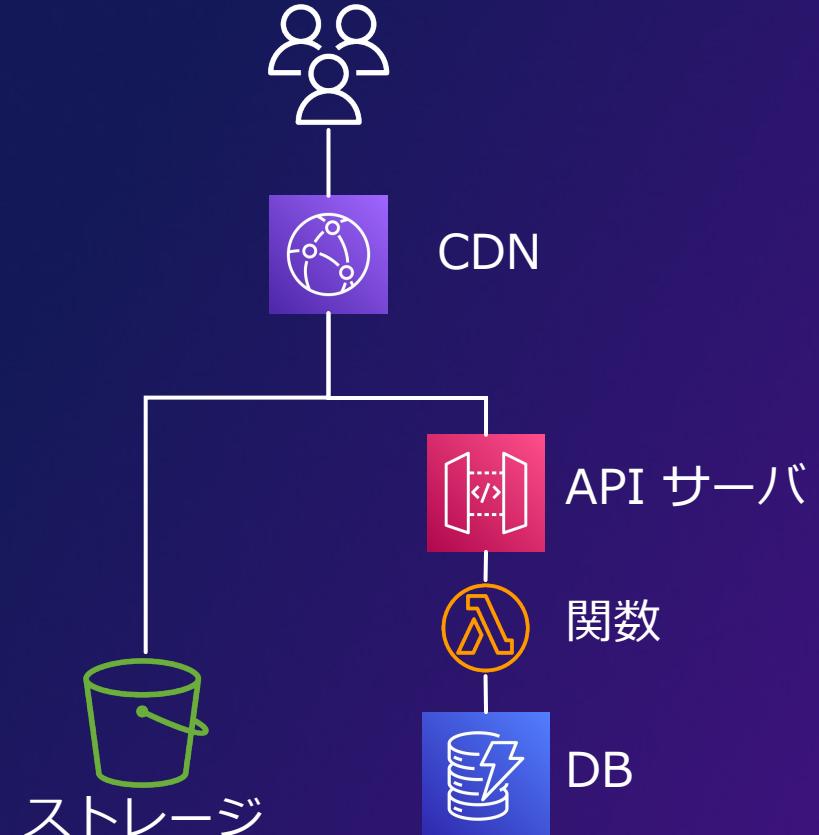


仮想マシン 仮想マシン



可用性：高

マネージド使用率：中



CDN



関数



DB

ストレージ

可用性：高

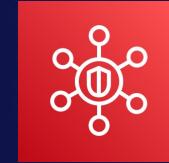
マネージド使用率：高

セキュリティ・構成管理サービスも様々提供



AWS Key Management Service (AWS KMS)

暗号鍵管理



AWS Security Hub

セキュリティダッシュボード



AWS Config

構成管理



AWS WAF

WAF



Amazon GuardDuty

機械学習を取り入れた脅威検知



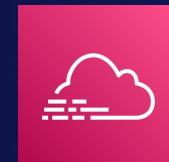
AWS Organizations

AWS アカウント統合管理



AWS Shield

DDoS対策



AWS CloudTrail

証跡ログ



AWS Control Tower

複数アカウントに対するガバナンス設定

クラウドのメリットをより活かすために



現在の構成を見直しながら、より良い環境にしていくために
ベンダー・職員間で認識の齟齬なくやりとりできるドキュメントがあることが望ましい

ドキュメントを使ってマルチテナント環境を統制



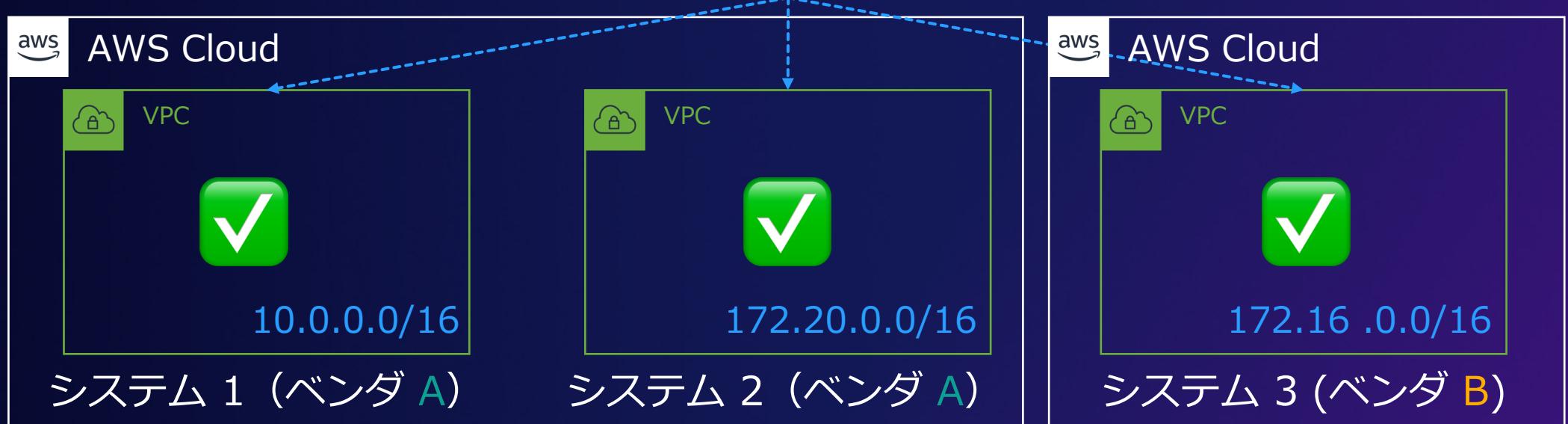
可用性



セキュリティ



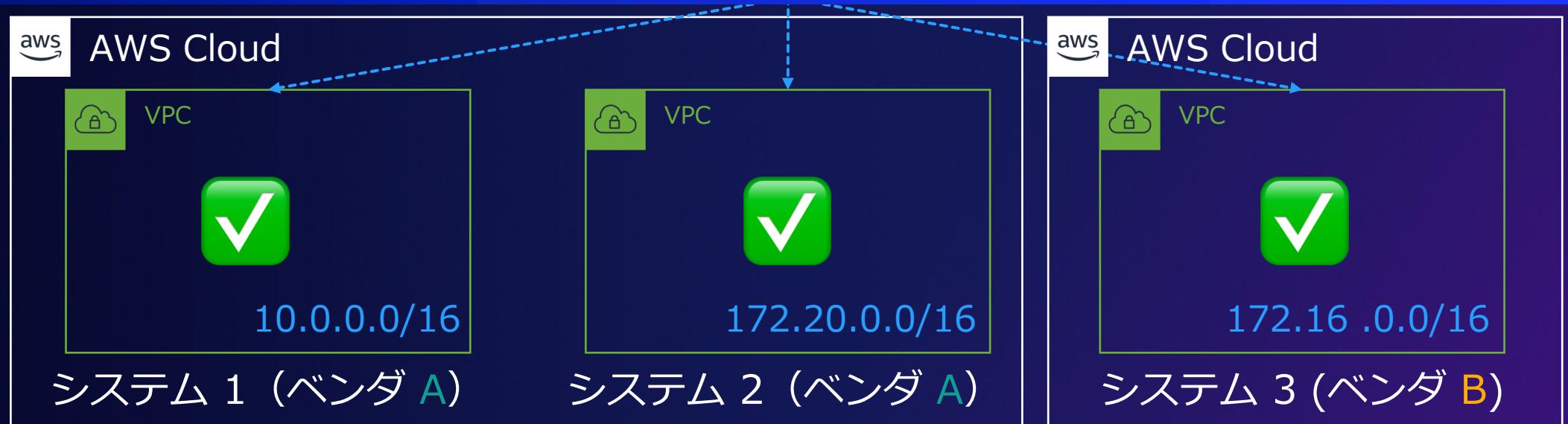
その他設定・構成



環境や対象が異なる様々なシステムに必要な
セキュリティポリシーやシステムの要件をドキュメントとして保存・共有し、
必要不可欠なサービス・機能の実装を促す

ドキュメントを使ってマルチテナント環境を統制

Infrastructure as Code

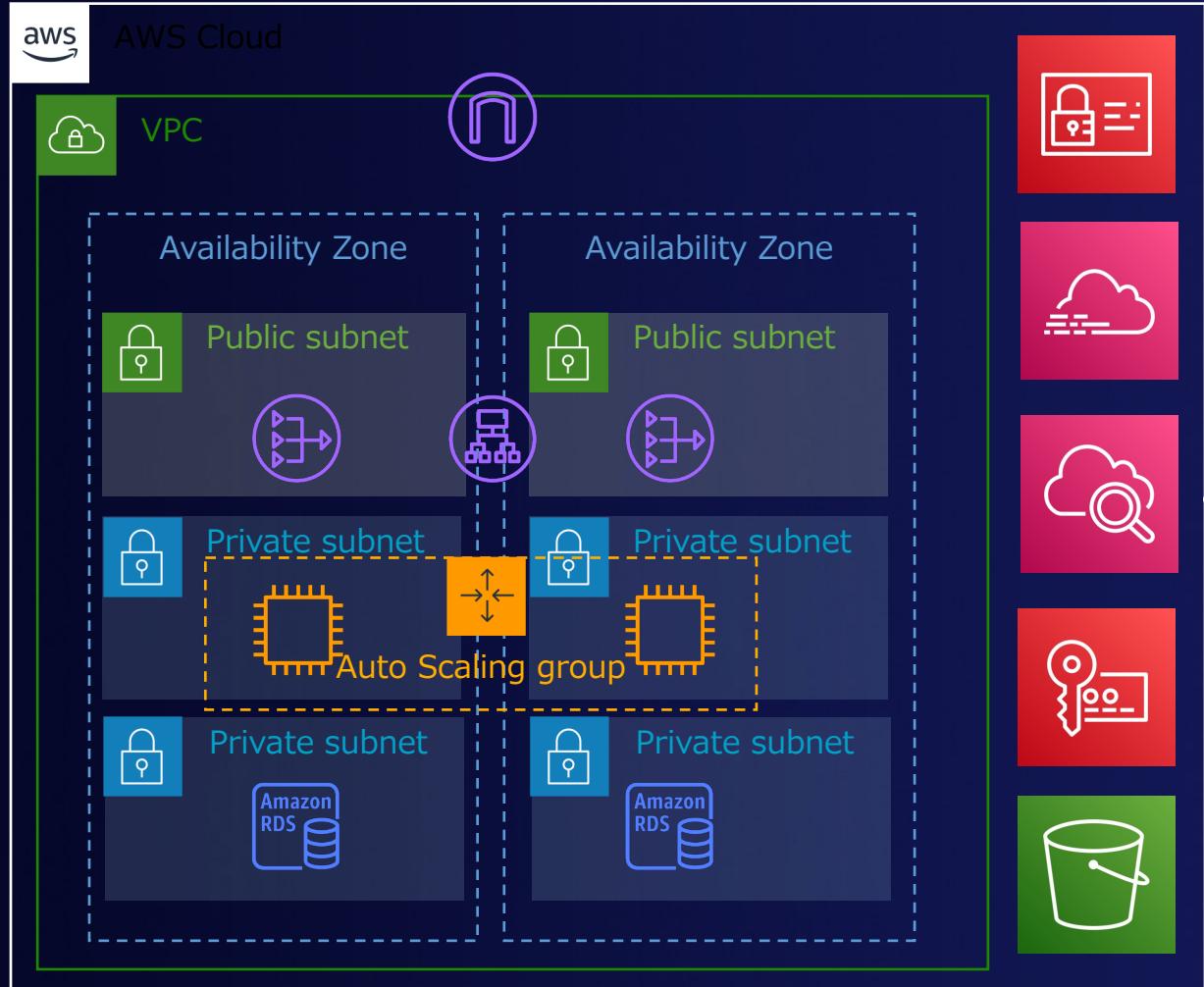


環境や対象が異なる様々なシステムに必要な
セキュリティポリシーやシステムの要件をドキュメントとして保存・共有し、
必要不可欠なサービス・機能の実装を促す

クラウド上の統制

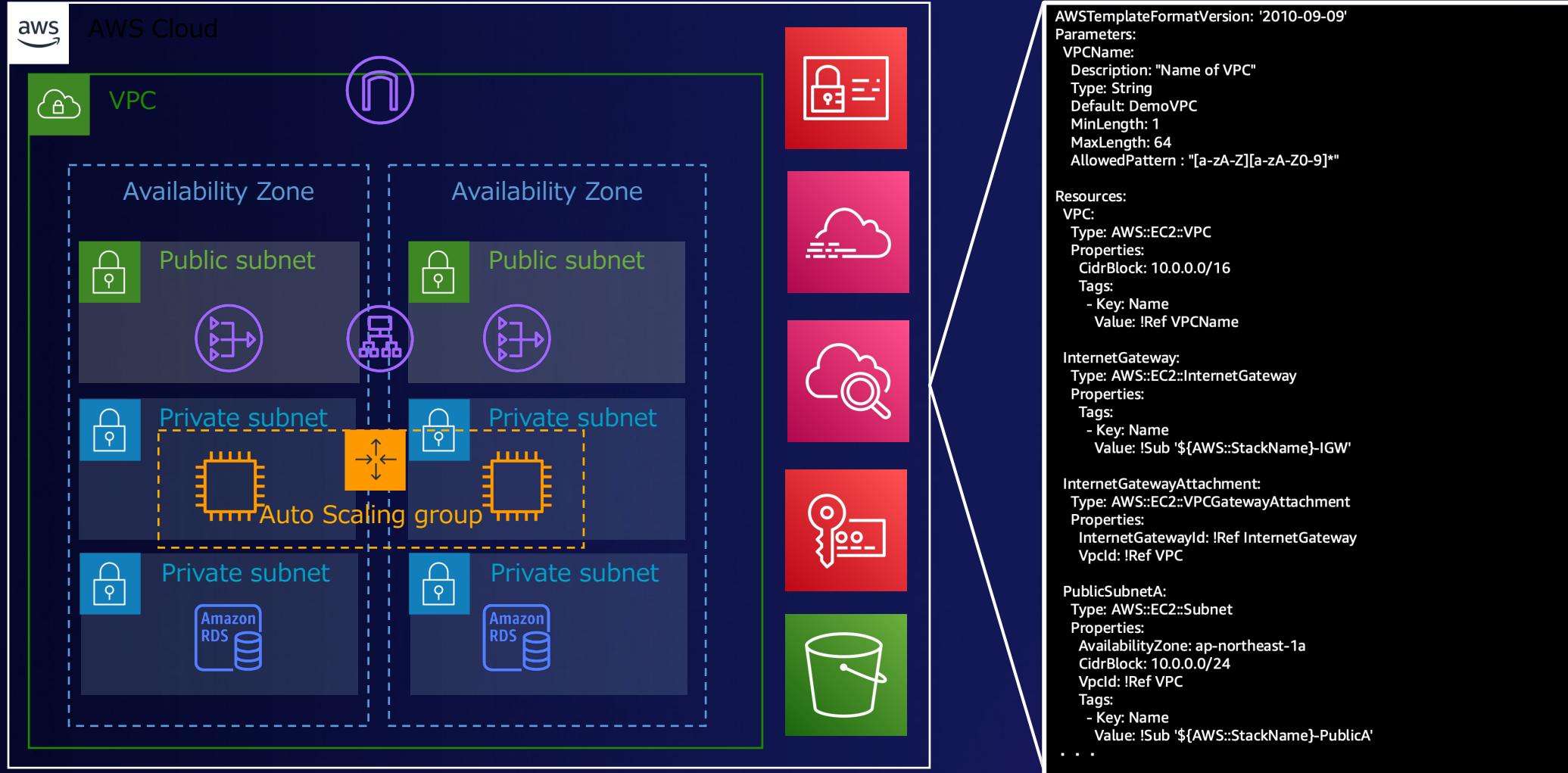
- IaC を使ってクラウド環境をあるべき状態に保つ -

Infrastructure as Code で「あるべき状態」を定義



インターネット接続：有
可用性レベル：同一リージョン・複数 AZ
自動スケール機能：ON
データ暗号化：あり
バックアップ：1日1回
ログイン時の2要素認証：あり
ログ収集：証跡ログ/アクセスログ
メトリクス収集：あり
・
・
・

Infrastructure as Code で「あるべき状態」を定義



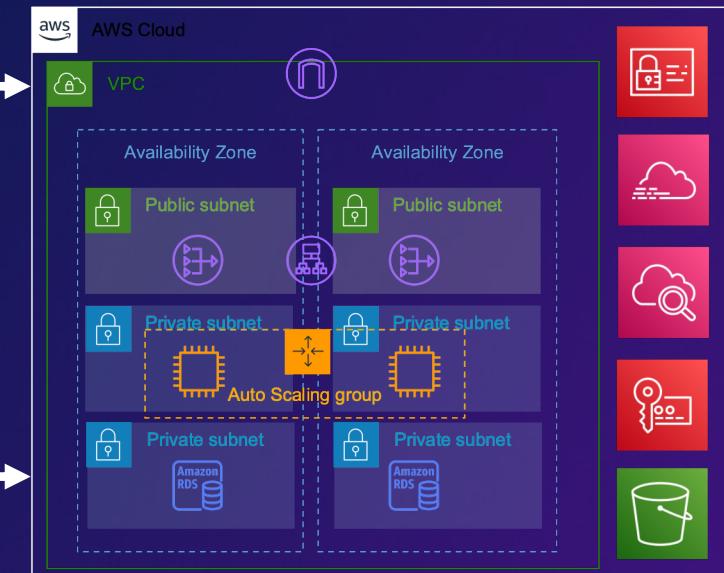
Infrastructure as Code (IaC) とは

IaC とは、コンピュータやソフトウェアの構成や設定に関する情報をプログラムコードとして記述し、専用のソフトウェアによって自動的に適用する手法。システム管理から手作業を減らして自動化、省力化を進め、安全性や安定性の向上にも資する。(IT用語辞典 <https://e-words.jp/w/IaC.html> より)

1. コードにて「るべき状態」を定義

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  VPCName:
    Description: 'Name of VPC'
    Type String
    Default: DemoVPC
    MinLength: 1
    MaxLength: 64
    AllowedPattern : "[a-zA-Z][a-zA-Z0-9]*"
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
        - Key: Name
          Value: !Ref VPCName
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-IGW'
  InternetGatewayAttachment:
    Type: AWS::EC2::VPCCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC
  PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: ap-northeast-1a
      CidrBlock: 10.0.0.0/24
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-PublicA'
...
```

2. ツールを通して自動的に環境を構築



3. 環境に変更が必要な場合はコードを変更

AWS における IaC 関連サービス



AWS CloudFormation

テキスト形式で宣言的に記述されたテンプレートから AWS リソースを作成



AWS Cloud Development Kit
(AWS CDK)

AWS の環境を一般のプログラミング言語で記述できるツールキット

バックエンドでは CloudFormation を利用

AWS における IaC 関連サービス

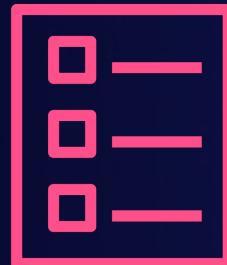


AWS CloudFormation

テキスト形式で宣言的に記述された
テンプレートから AWS リソースを作成

AWS CloudFormation を利用したサービスの設定

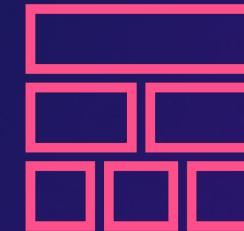
テンプレート



AWS CloudFormation



スタック



JSON/YAML 形式のテキスト

パラメータの定義
リソースの作成
実際の設定

フレームワーク

スタックの作成
スタックの更新
エラー検知とロールバック

AWS サービスの設定

サービス全体での統合
サービスのイベント管理
カスタマイズ

テンプレート

- AWS CloudFormation の心臓部
- スタック構築の設計図
 - どのリソースをどう起動するかがすべて記述されている
 - Resource の依存関係は CloudFormation が自動判別
- JSON/YAML フォーマットで記述

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  VPCName:
    Description: "Name of VPC"
    Type: String
    Default: DemoVPC
    MinLength: 1
    MaxLength: 64
    AllowedPattern : "[a-zA-Z][a-zA-Z0-9]*"

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
        - Key: Name
          Value: !Ref VPCName

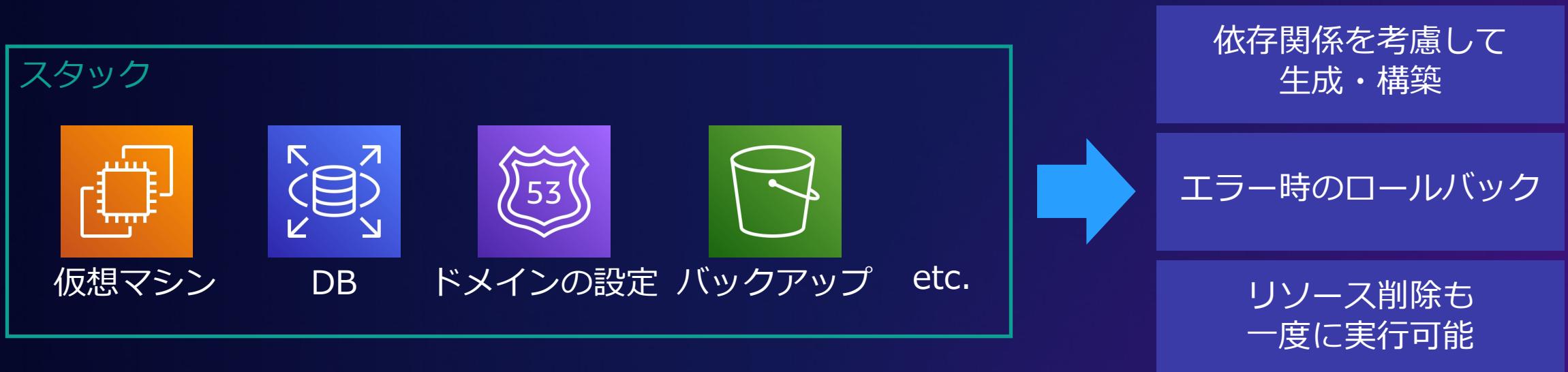
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-IGW'

  InternetGatewayAttachment:
    Type: AWS::EC2::VPGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC

  PublicSubnetA:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: ap-northeast-1a
      CidrBlock: 10.0.0.0/24
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-PublicA'
      . . .
```

スタック

- テンプレートからプロビジョニングされるリソースの集合のこと
- スタック単位でリソースの管理が可能。スタック破棄を実行すると、スタックにひもづくリソースを破棄することが可能
- 使用するリソースおよびリソースの構築順は、テンプレートの依存関係から CloudFormation が自動的に決定



AWS における IaC 関連サービス (再掲)



AWS Cloud Development Kit (AWS CDK)

AWS の環境を一般のプログラミング言語で
記述できるツールキット

バックエンドでは CloudFormation を利用

AWS Cloud Development Kit (CDK)



一般のプログラミング言語を利用して AWS スタックを定義可能

CDK を利用すると変わること

CloudFormation

```
AWSTemplateFormatVersion: '2010-09-09'  
Parameters:  
  VPCName:  
    Description: "Name of VPC"  
    Type: String  
    Default: DemoVPC  
    MinLength: 1  
    MaxLength: 64  
    AllowedPattern : "[a-zA-Z][a-zA-Z0-9]*"  
  
Resources:  
  VPC:  
    Type: AWS::EC2::VPC  
    Properties:  
      CidrBlock: 10.0.0.0/16  
      Tags:  
        - Key: Name  
          Value: !Ref VPCName  
  
  InternetGateway:  
    Type: AWS::EC2::InternetGateway  
    Properties:  
      Tags:  
        - Key: Name  
          Value: !Sub '${AWS::StackName}-IGW'  
....
```

YAML, JSON のリストで定義

Cloud Development Kit

```
1 import { Construct } from "@aws-cdk/core";  
2 import { Bucket, BucketEncryption, BucketProps } from "@aws-cdk/aws-s3";  
3  
4 export class EncryptedBucket extends Construct {  
5   constructor(scope: Construct, id: string, props?: BucketProps) {  
6     super(scope, id);  
7  
8     let newProps: BucketProps = { ...props };  
9     if (  
10       !props ||  
11       props?.encryption === undefined ||  
12       props?.encryption === BucketEncryption.UNENCRYPTED  
13     ) {  
14       newProps.encryption = BucketEncryption.KMS_MANAGED;  
15     }  
16     new Bucket(this, `${id}-bucket`, newProps);  
17   }  
18 }  
19
```

条件分岐や定義の繰り返しを
使用可能

手作業より自動化しやすい形で環境を定義

手作業を記録する場合

Word や Excel 等ドキュメントで手順を管理する場合、変更履歴の保存が必要

手順が正しいか時間をかけて確認する必要がある

特に本番環境の場合、作業者を増やして作業を確認し合うなど、正しくデプロイする工夫が必要

作業ミスが発生した場合の対応やロールバックの手法についても検討が必要

•
•
•



IaC の場合

コードで定義した内容を Git で管理する場合、最新版や変更履歴を把握しやすい

コードを実行すればデプロイや整合性の確認が自動的に実行されるため、テスト時間を短縮

テストが正しく行われていれば、同じコードを使うことで正しくデプロイされることが保証

コードに問題があった場合、自動的に問題を検出しロールバックを実行

•
•
•

Infrastructure as Code のメリット

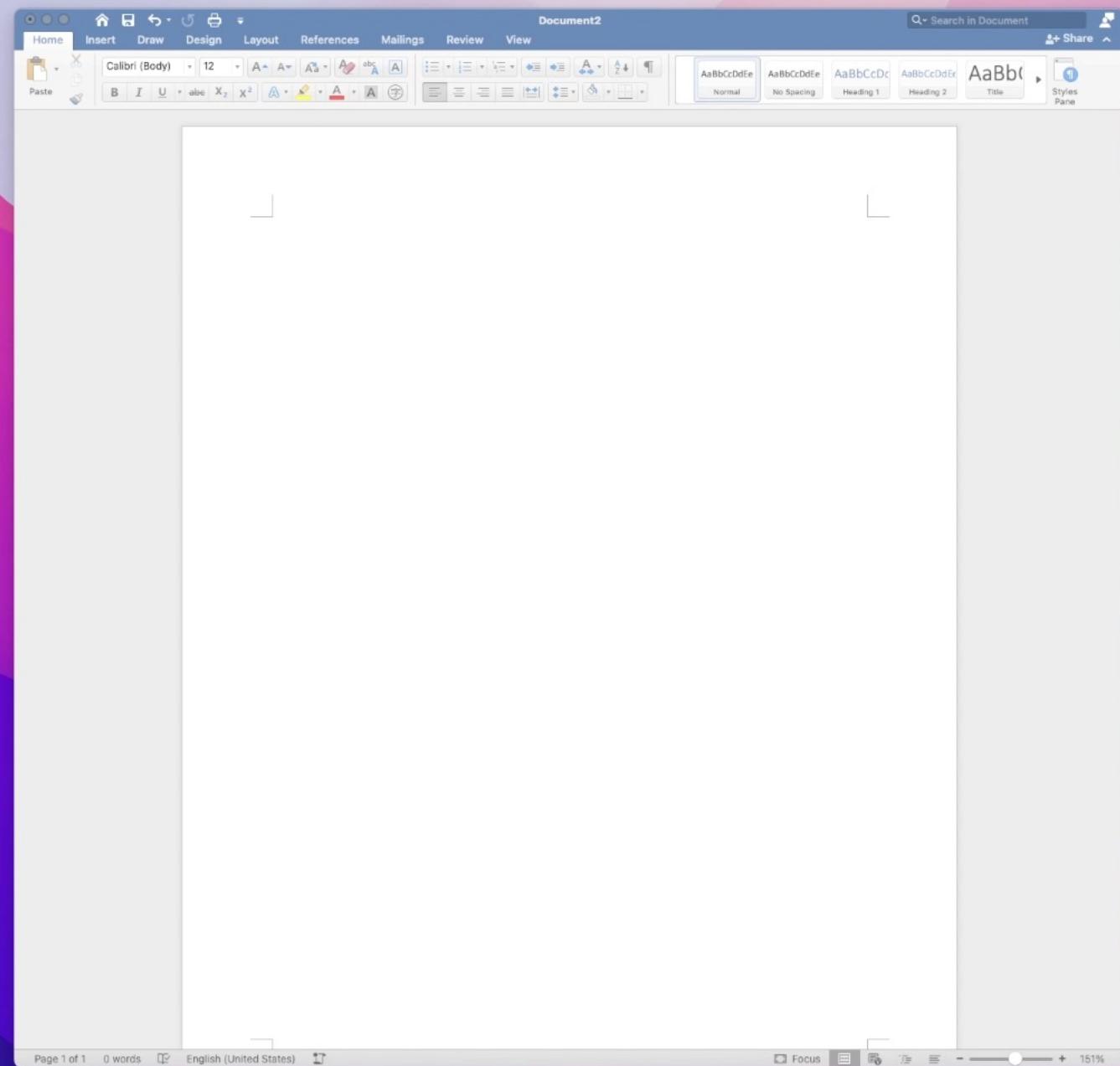
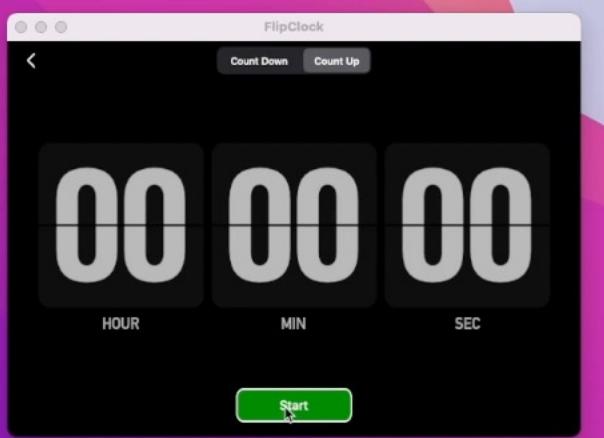


Demo

- ・マネジメントコンソールの操作から手順書の作成
- ・同じ環境を CDK を使用して作成

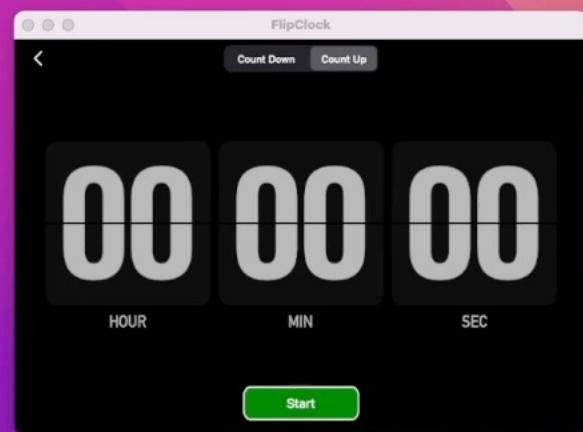
Demo

- ・マネジメントコンソールの操作から手順書の作成
- ・同じ環境を CDK を使用して作成



Demo

- ・マネジメントコンソールの操作から手順書の作成
- ・同じ環境を CDK を使用して作成



app-stack.ts — SummitVideo

TS app-stack.ts ●

CDK > app > lib > **TS** app-stack.ts

1 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

zsh - app + ↻ ✎ ×

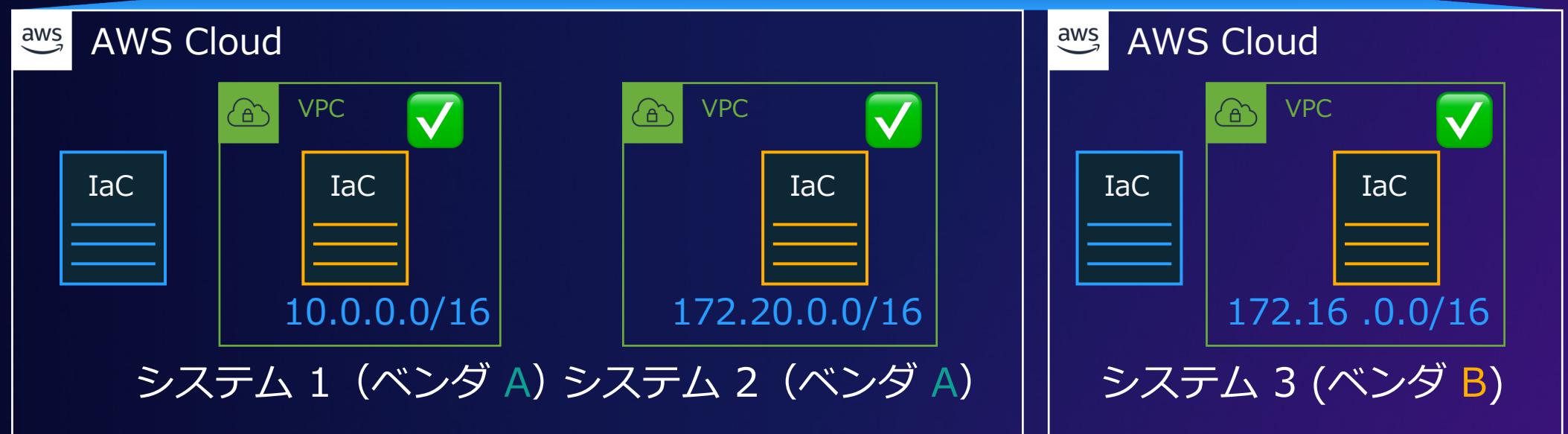
home Aa ab * ↑ ↓ ×

✓ Environment aws://186137372821/ap-northeast-1 bootstrapped.

tabukimt@3c22fbccdfdd app %
tabukimt@3c22fbccdfdd app %
tabukimt@3c22fbccdfdd app %

master* ↻ ✎ 0 ▲ 0 AWS: profile:default Ln 1, Col 1 Spaces: 2 UTF-8 LF {} TypeScript ⚡ 🔍

IaC 製のテンプレートでマルチテナント環境を統制



IaC を使用したテンプレートを活用することで、システムに対して必要な設定を事前に定義

テンプレートを使った統制

- Baseline Environment on AWS -

Baseline Environment on AWS (BLEA)

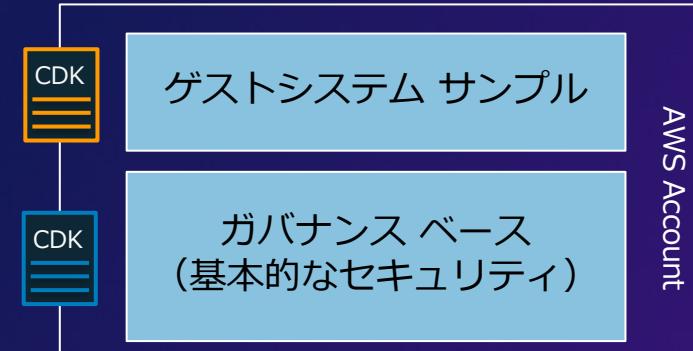
<https://github.com/aws-samples/baseline-environment-on-aws>

AWS のセキュリティベストプラクティスを実装した
サンプルテンプレート

特徴

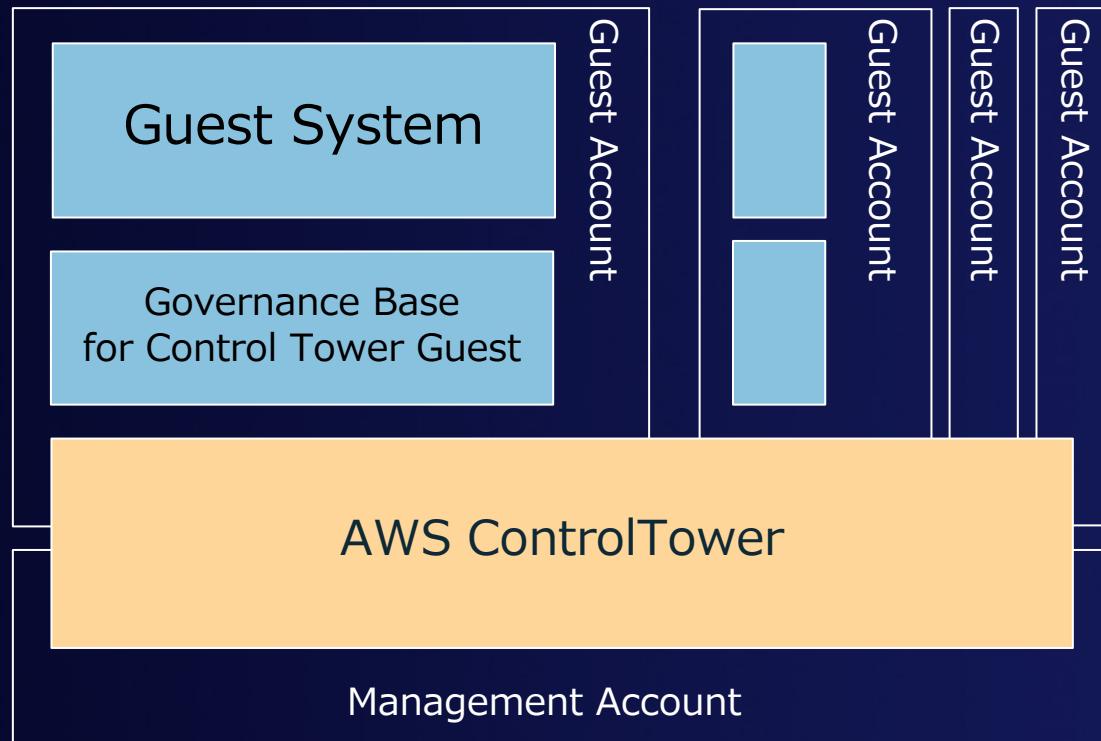
- 基本的なセキュリティを設定するテンプレートと
ゲストシステムのサンプルテンプレートを提供
- AWS のセキュリティベストプラクティスに準拠
- Cloud Development Kit (CDK) コード
参考となるスニペット、コメント、リファレンスを豊富に記載
- チームによる長期的な利用を想定
CDK 標準ライブラリのみを使ったシンプルな実装
利用者が理解しやすいよう過度な作り込みを避ける

BLEAが提供するテンプレート



BLEA の 2 つのアカウント管理パターン

マルチアカウント版



シングルアカウント版
(Standalone)



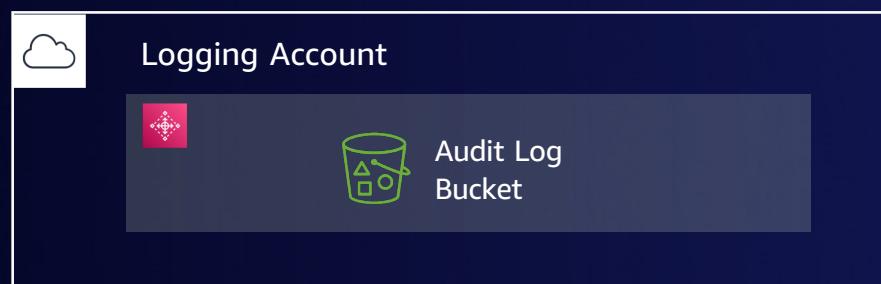
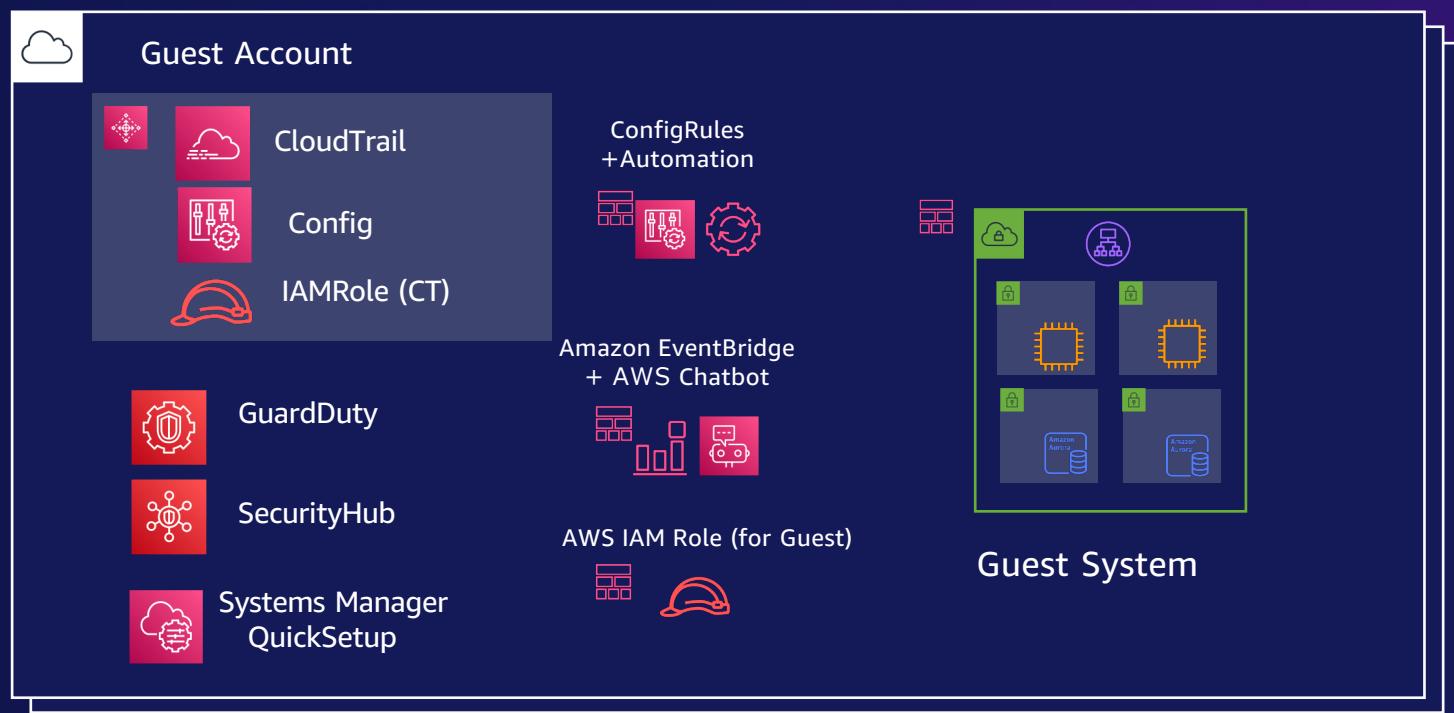
■ Baseline Environment on AWS の提供範囲

※Governance Base によって実現されるセキュリティは、マルチアカウント版もシングルアカウント版も同じ

※“Guest system” はマルチアカウント版もシングルアカウント版も同じものが利用可能

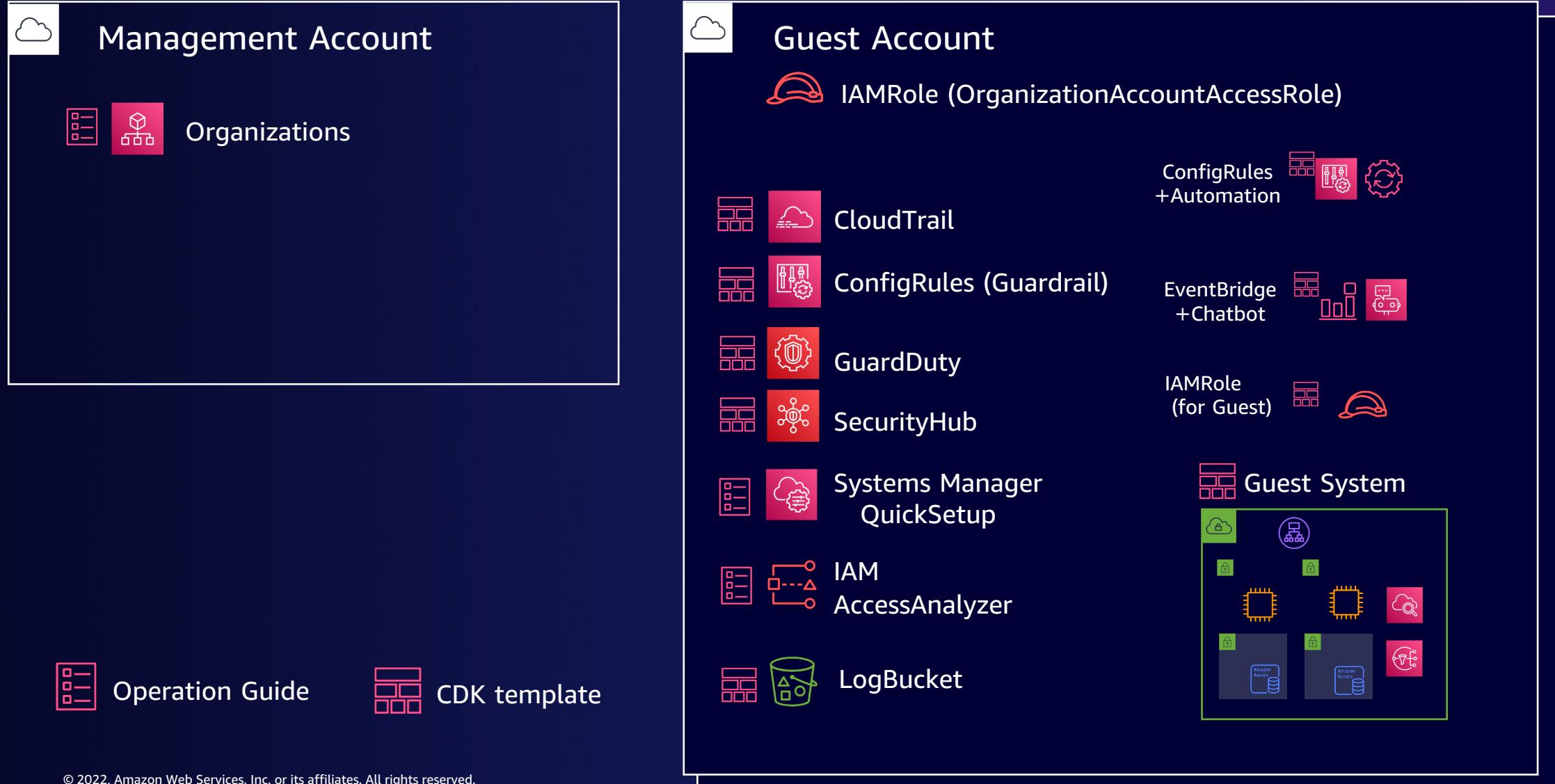
BLEA マルチアカウント版 – 利用サービス

(ver. 2021/10/26)



BLEA シングルアカウント版 – 利用サービス

(ver. 2021/10/26)



ガバナンスの全体像 - BLEA マルチアカウント版

(ver. 2021/10/26)

AWS ControlTower

Governance Base for CT Guest

Guest System



	管理タスク	Management Account	Audit Account	LogArchive Account	Shared Svc Account*	Guest Account
AWS ControlTower	1 アカウント払い出し	CT-Org	(Created by CT)	(Created by CT)	(Created by CT)	(Created by CT)
	2 アクセス制御	CT-SSO/Admin + AD	CT-Admin	CT-Admin	CT-Admin	CT-Admin
	3 予防的統制	CT-SCP	CT-SCP	CT-SCP	CT-SCP	CT-SCP
	4 発見的統制 (Config)	CT-ConfigRules 作成	CT-ConfigRules	CT-ConfigRules	CT-ConfigRules	CT-ConfigRules
	5 ロギング	(CT-Log Bucket)	CT-CloudTrail/Config	CT-Log Bucket	CT-CloudTrail/Config	CT-CloudTrail/Config
	6 通知 (CT)	(CT-Audit Topic)	CT-Audit Topic	(To Audit Topic)	(To Audit Topic)	(To Audit Topic)
	7 発見的統制 (挙動)		MNL-SecurityHub MNL-GuardDuty			Member-SecurityHub Member-GuardDuty
	8 セキュリティ分析		MNL-IAM-AccessAnalyzer			Member-IAMAccessAnalyzer
	9 共有ネットワーク				TMPL-VPC/DNS/VPCEP *	(Use Shared Svc Account)
	10 サーバ管理					MNL-SSM QuickSetup
	11 通知 (Security) + Chat					TMPL-Security Alarm
	12 アクセス制御 (for Guest)					TMPL-IAM
	13 発見的統制 (for Guest)					TMPL-ConfigRules
	14 ロギング (for Guest)					TMPL-FlowLogs/ALB Logs etc.
	15 ネットワーク (for Guest)					TMPL-VPC
	16 鍵管理 (for Guest)					TMPL-KMS
	17 通知 (Monitoring) + Chat					TMPL-Monitor Alarm
	18 リソース + バックアップ					TMPL-EC2/Serverless etc.
	19 デプロイメント					TMPL-CI/CD

ガバナンスの全体像 - BLEA シングルアカウント版

(ver. 2021/10/26)

Governance Base for Standalone Guest

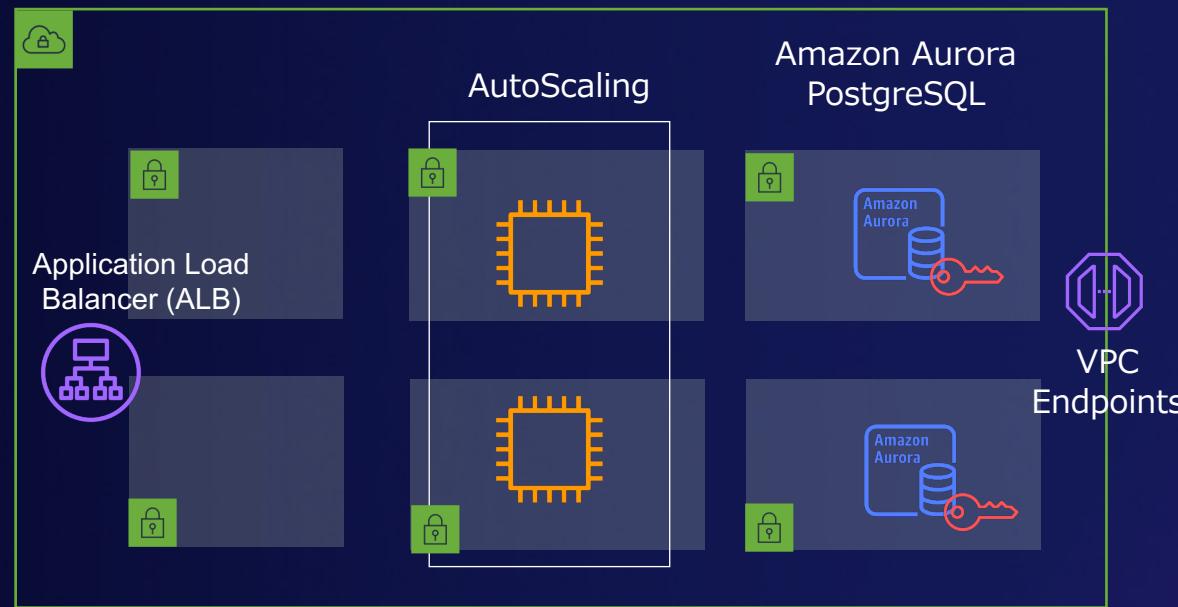
Guest System



	管理タスク	Management Account	Audit Account	LogArchive Account	Shared Svc Account	Guest Account
1	アカウント払い出し					Manual-Organizations
2	アクセス制御					(Organizations Roles)
3	予防的統制					(MNL-IAM)
4	発見的統制 (Config)					TMPL-ConfigRules
5	ログイン					TMPL-CloudTrail/Config
6	通知 (CT)					(None)
7	発見的統制 (挙動)					TMPL-SecurityHub TMPL-GuardDuty
8	セキュリティ分析					IAM-AccessAnalyzer
9	共有ネットワーク					(None)
10	サーバ管理					MNL-SSM QuickSetup
11	通知 (Security) + Chat					TMPL-Security Alarm
12	アクセス制御 (for Guest)					TMPL-IAM
13	発見的統制 (for Guest)					TMPL-ConfigRules
14	ログイン (for Guest)					TMPL-FlowLogs/ALB Logs etc.
15	ネットワーク (for Guest)					TMPL-VPC
16	鍵管理 (for Guest)					TMPL-KMS
17	通知 (Monitoring) + Chat					TMPL-Monitor Alarm
18	リソース + バックアップ					TMPL-EC2/Serverless etc.
19	デプロイメント					TMPL-CI/CD

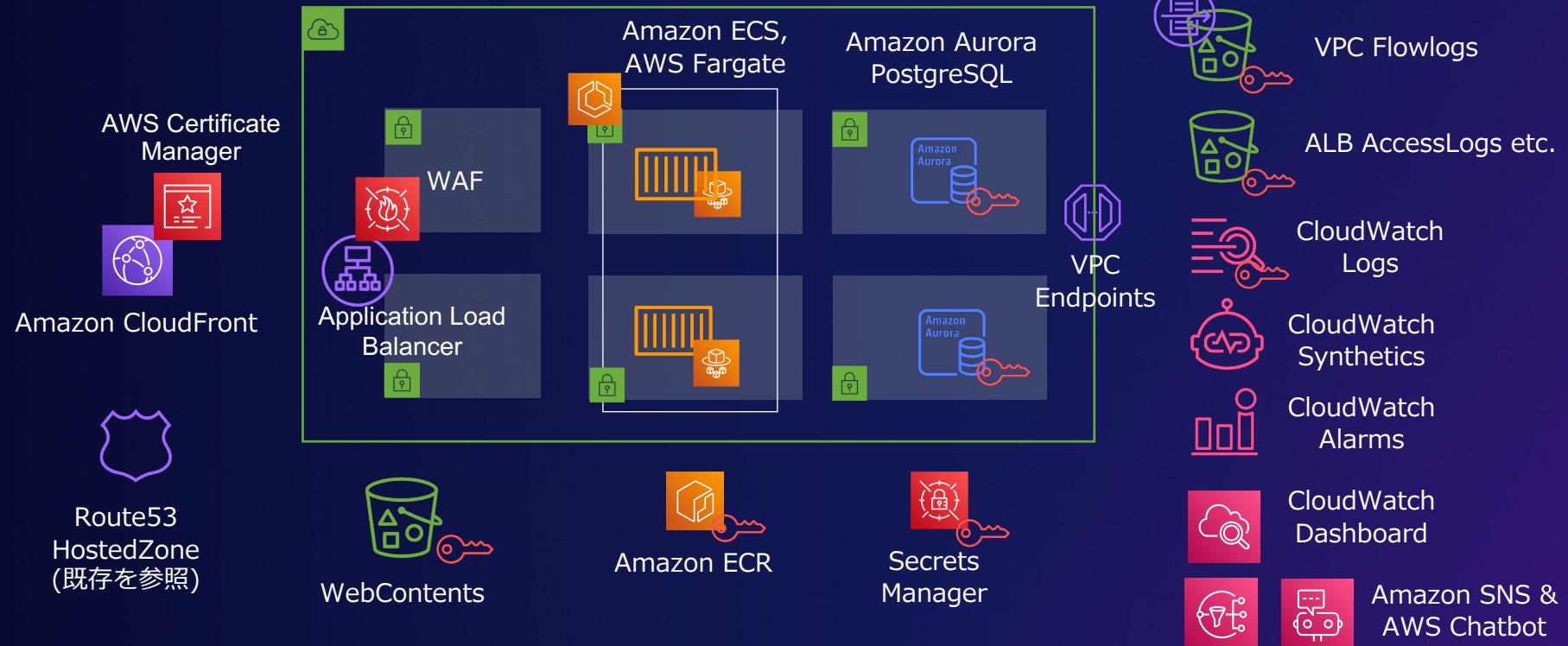
ゲストシステムサンプルコード 1： Web アプリケーション(仮想マシン(EC2)を使用)

(ver. 2021/10/26)



ゲストシステムサンプルコード2： Web アプリケーション(コンテナを使用)

(ver. 2021/10/26)



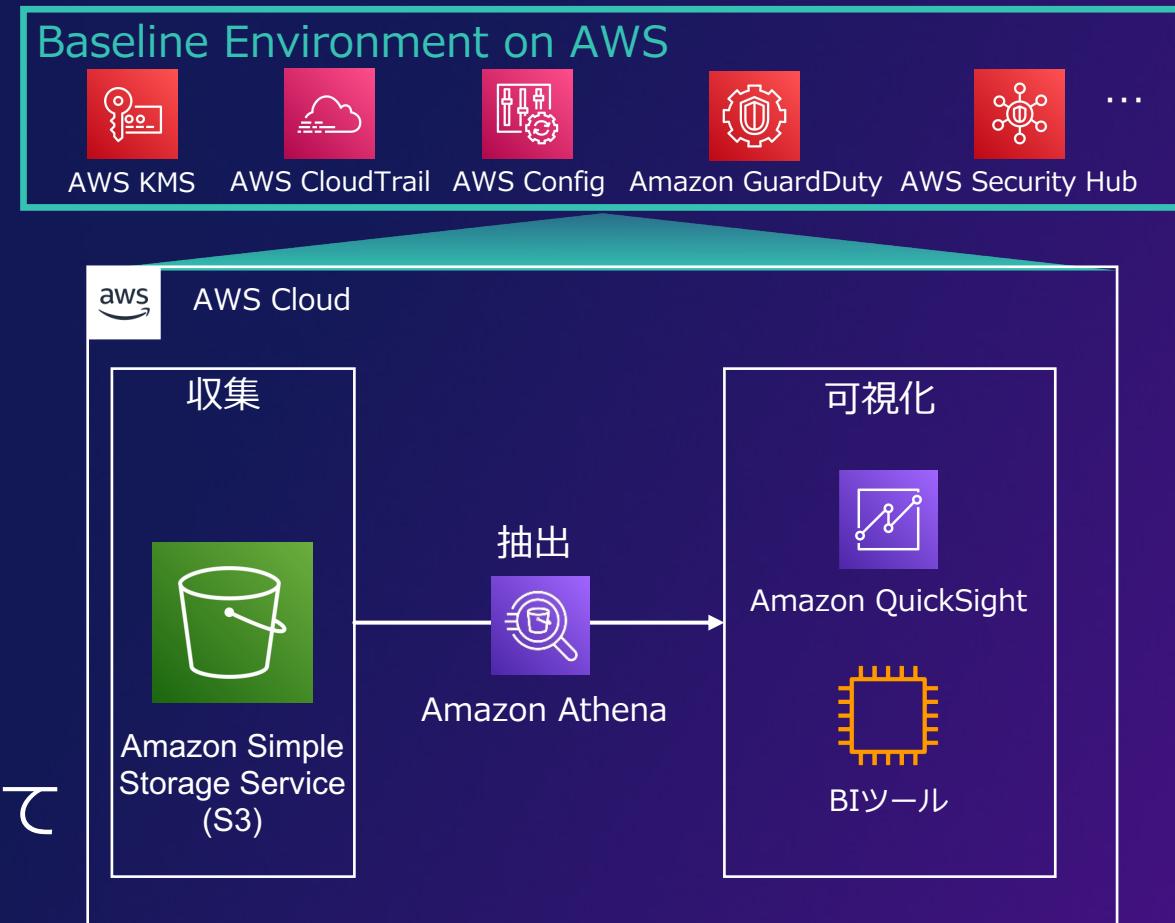
神戸市様： AWS 上のデータレイク環境に BLEA を適用

自治体観点の効果：

- AWS のセキュリティーの
ベストプラクティスに沿った設定を
迅速に適用
- 自庁に合わせた構成・設定に
カスタマイズ

開発者観点の効果：

- 周辺技術・サービス選定の簡略化
- サービス間連携を CDK で行う方法について
BLEA を通して理解度が向上



DEMO

- BLEA におけるガバナンスベースを展開



サービス

サービス、特徴、ブログ、およびドキュメントなどを検索

[オプション+S]



東京 ▾

TeamRole/MasterKey @ 4055-3827-4237 ▾

GuardDuty

Security Hub

Config

S3

EC2

RDS

コンソールのホーム 情報

アクション ▾

最近アクセスしたサービス 情報

S3



Config



Security Hub



GuardDuty



Cloud9



EC2



CloudTrail



Simple Notification Service



Lambda



Directory Service



WorkSpaces



Amazon WorkDocs



EC2 Image Builder



CloudFront



VPC

AWS へようこそ



AWS の開始方法

AWS を最大限に活用するために基礎を学び、有益な情報をを見つけましょう。



トレーニングと認定

AWS のエキスパートから学び、スキルと知識を深めましょう。



AWS の最新情報

新しい AWS のサービス、機能、およびリージョンについてご覧ください。

AWS Health 情報コストと使用状況 情報

今月のコスト

\$0.00

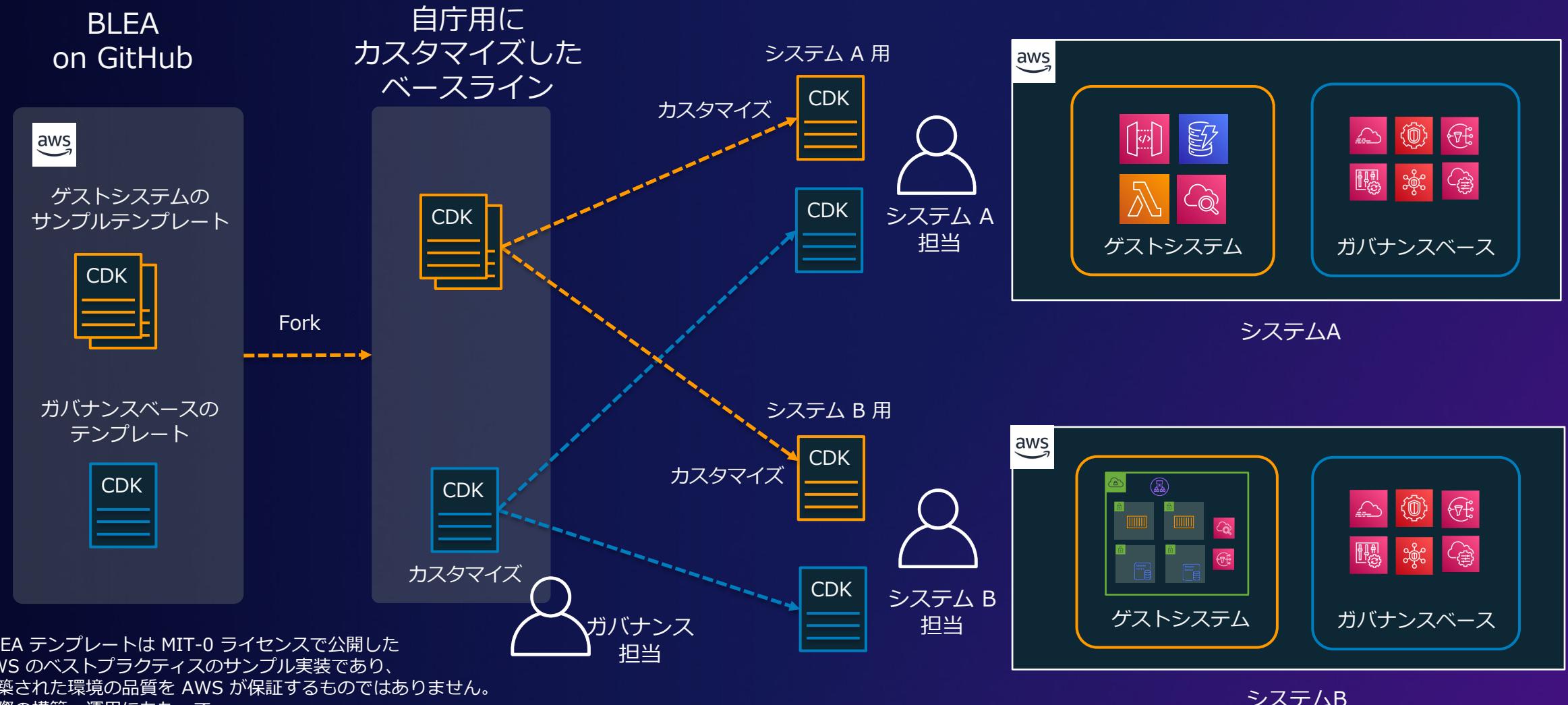
予測される月末のコスト

\$0.00

今月の最高コスト



Baseline Environment on AWS の利用方法



まとめ

- ・ 公共機関のお客さまにおいては、マルチベンダーや関連組織の統合といった観点から、マルチテナント環境における統制を求められる
- ・ こうしたマルチテナント環境において、セキュリティや可用性の面からクラウド利用の指針「あるべき状態」を定義することが重要
- ・ Infrastructure as Code を利用することであるべき状態を事前定義しマルチテナント環境を統制することが可能
- ・ ガバナンスおよびシステムのサンプルテンプレートである Baseline Environment on AWS (BLEA) をオープンソースとして公開

今から始めるために

- AWS の IaC に関するサービス・ソリューションの情報は以下から入手可能です。ぜひ実際のクラウド環境にご利用ください
 - CloudFormation ハンズオン
 - https://pages.awscloud.com/JAPAN-event-OE-Hands-on-for-Beginners-cfn-2020-reg-event-LP.html?trk=aws_introduction_page
 - CDK ベストプラクティス
 - <https://aws.amazon.com/jp/blogs/news/best-practices-for-developing-cloud-applications-with-aws-cdk/>
 - Baseline Environment on AWS
 - <https://github.com/aws-samples/baseline-environment-on-aws>

あわせてご視聴ください：

セッション番号：AWS-22

「テンプレートによるAWS環境のガバナンス

- Baseline Environment on AWS (BLEA) 徹底解説 -」

Thank you!

Masahiro Tabuki



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.