

AWS DevOps 祭り 2018

# AWS Management Tools サービスアップデートのご紹介

大 村幸敬  
ソリューションアーキテクト  
アマゾン ウェブ サービス ジャパン株式会社



# 大村 幸敬 (おおむら ゆきたか)

ソリューションアーキテクト

- お客様のパートナーとして  
AWSに限らず最適なアーキテクチャ検討をサポート
- エンタープライズ企業を担当
- Management Tools & DevOps 系サービス担当

Background : SI / Infra / Financial / DevOps

# アジェンダ

1. DevOps と AWS Management Tools
2. AWS Management Toolsの概要
3. 各サービスの使い所と直近のアップデート
4. マルチアカウント管理

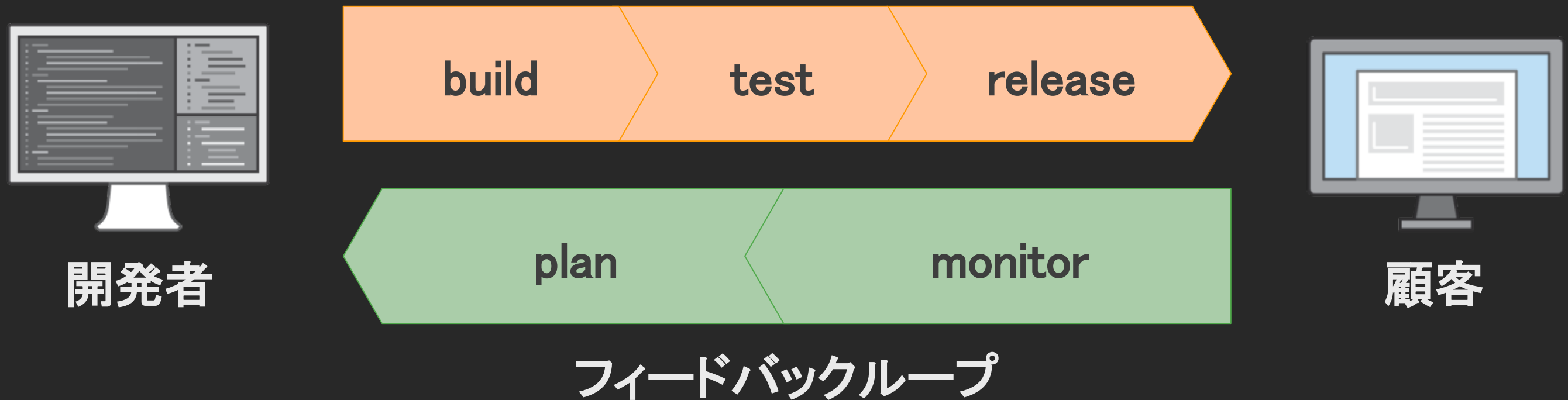
# Notice

- 直近3ヶ月からre:Invent 2018 までのアップデートを中心に扱います

# DevOps と AWS Management Tools

# What is DevOps?

## デリバリのパイプライン



DevOps = 無駄やボトルネックを取り除くことで、  
ライフサイクルを効率化し、高速化すること

# Metrics for DevOps

作り始めてからデプロイされるまでのストリームが重要  
ストリームのベロシティ(速度)をどう改善するか

企画

見積もり

承認

インフラ調達

設計

環境構築

開発

ビルド

テスト

Provision

デプロイ

監視

運用



# ベロシティ向上の阻害要因



心配

責任



# ベロシティ向上のために自動化を

- 不安を取り除くためにシステムを使う
  - 正しくOperationが行えるスクリプト
  - 問題があったときに止められるシステム
  - バグがあったときに再発防止できるテストコード



# 全てを自動化する

自動化は最適化を可能にする

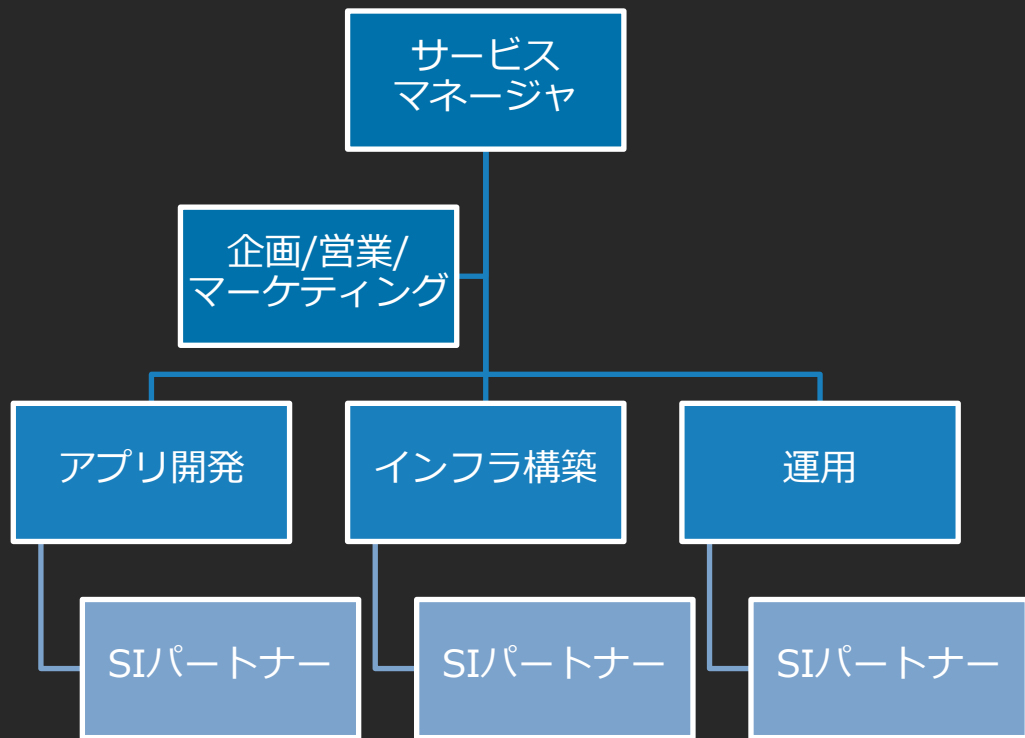
- “どのような手順も明確にされ、自動化することは可能”  
– Werner Vogels

インフラ周りの新しいマインドセット

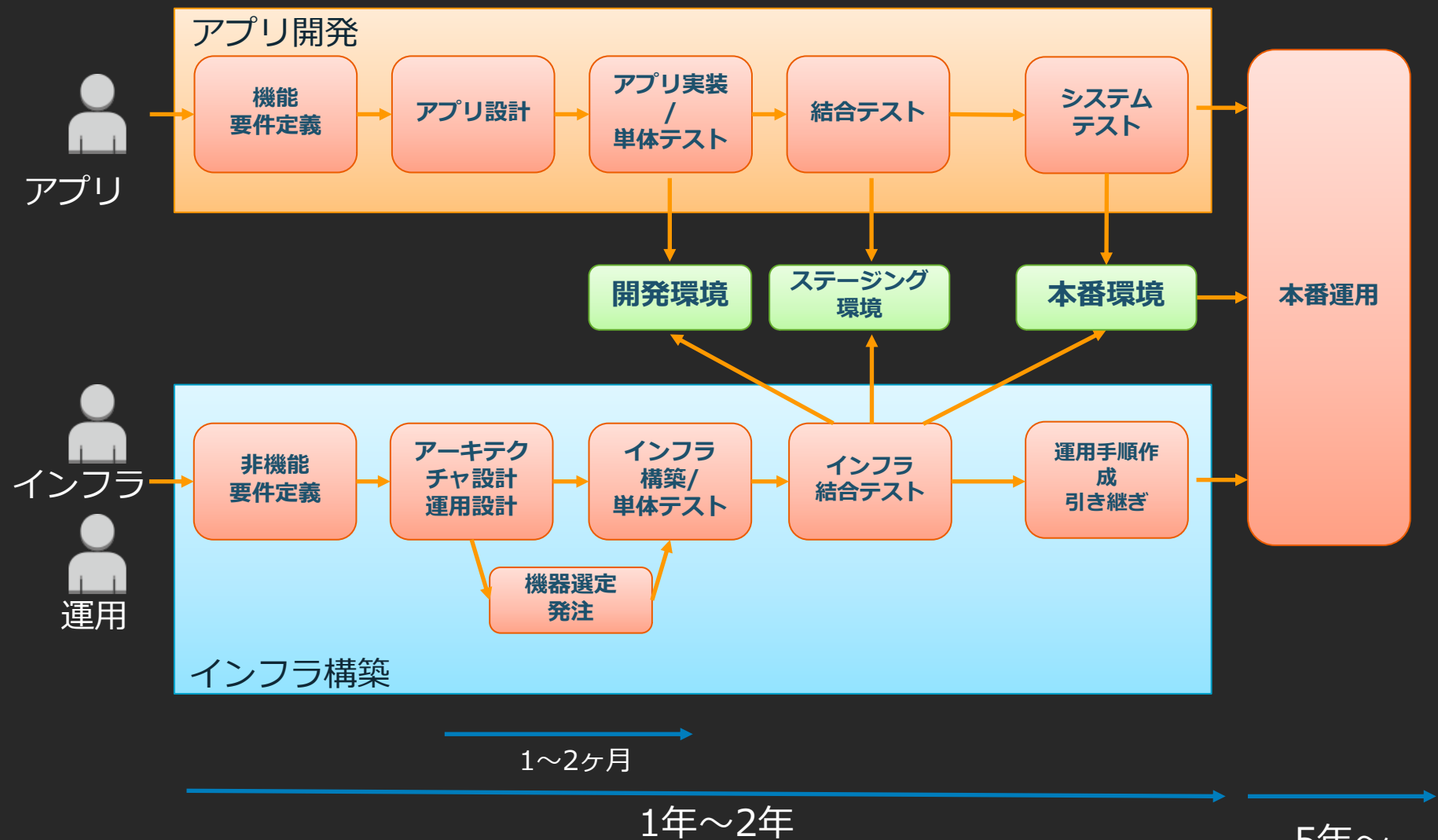
- 環境に関するすべてのことはコードで説明される
- 自動化はサービスによってもたらされる
- 自動化はツールとフレームワークの要件を導出する
- 頻繁なリリースは、頻繁なイノベーションを起こす

# オンプレミスでの開発の一般的な体制とスケジュール

## 開発・運用体制

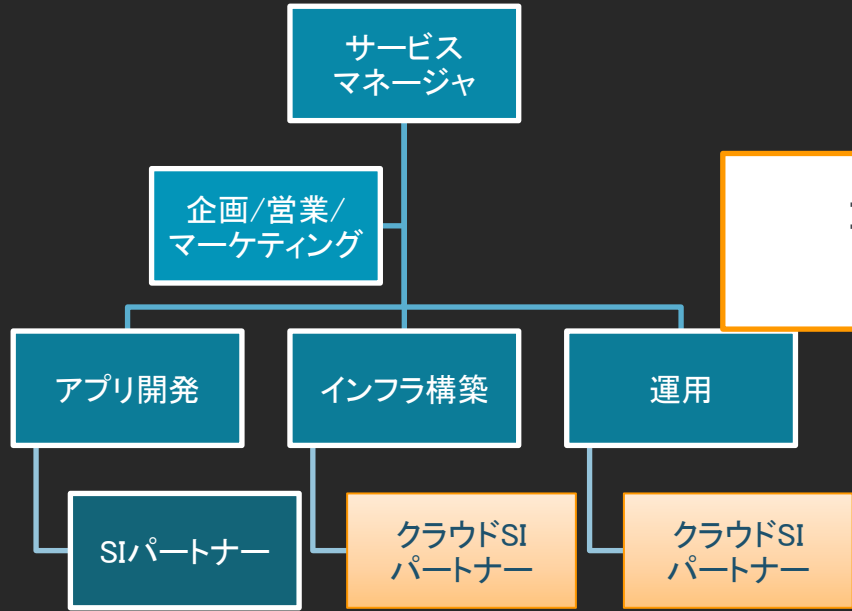


## 開発スケジュール



# クラウドでの開発の一般的な体制とスケジュール(既存踏襲)

## 開発・運用体制



アプリ

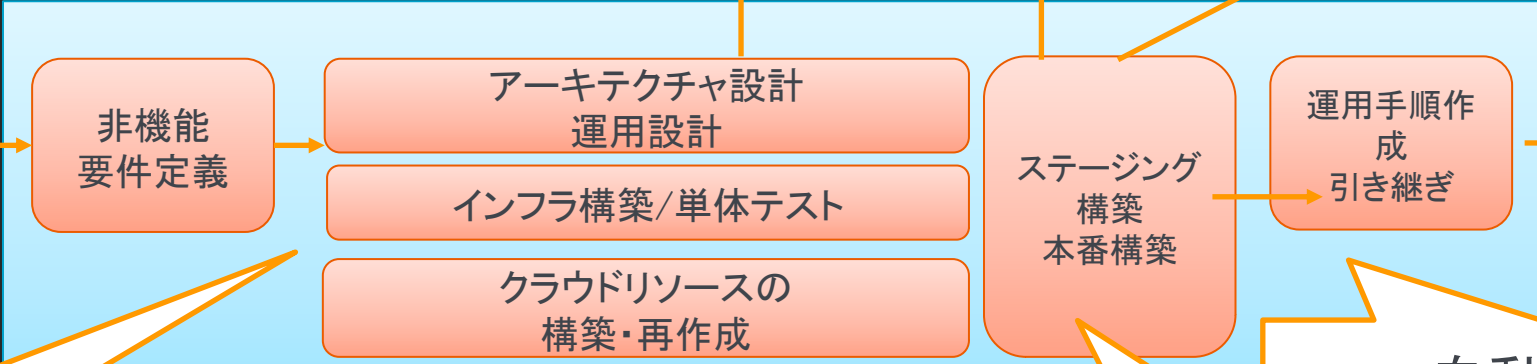


アプリ開発自体は大きく変わらない

コミュニケーション量はこれまでと変わらない



インフラ  
運用



設計・実装の深さが浅く

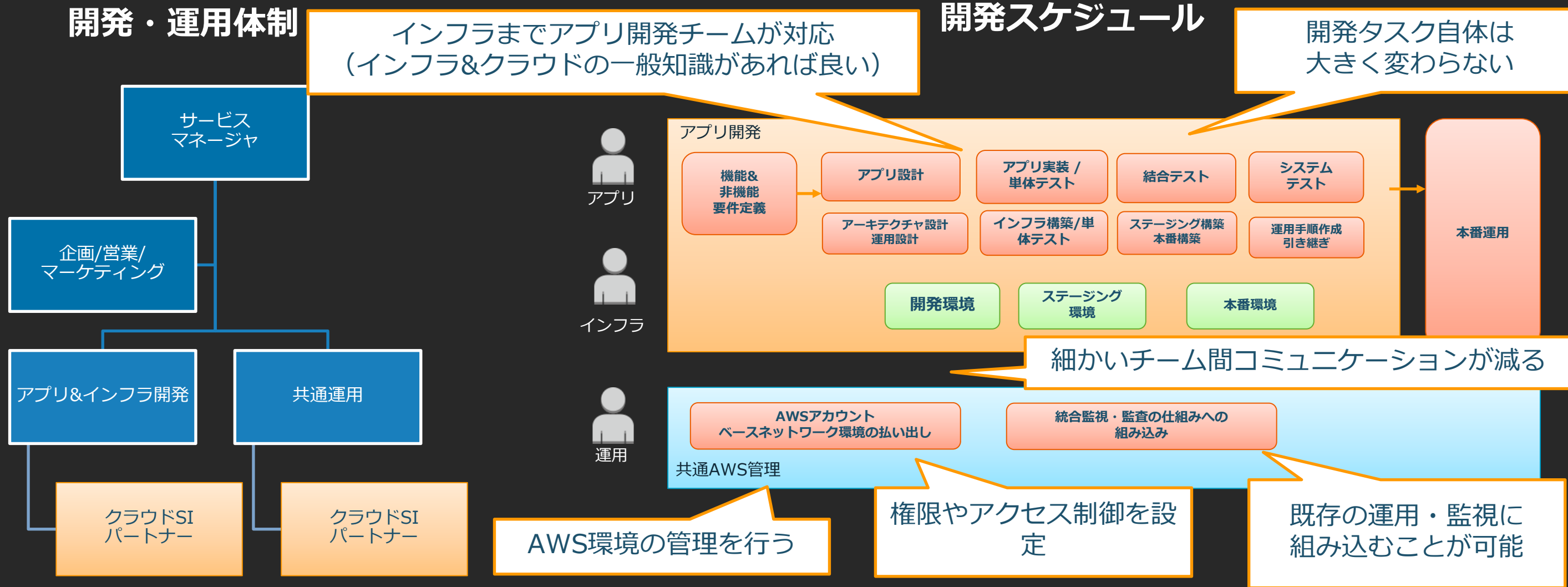
自動化による運用省力化が可能

1~2ヶ月 → 数分  
大規模投資 → 初期投資不要

本番・ステージングは自動構築も可能

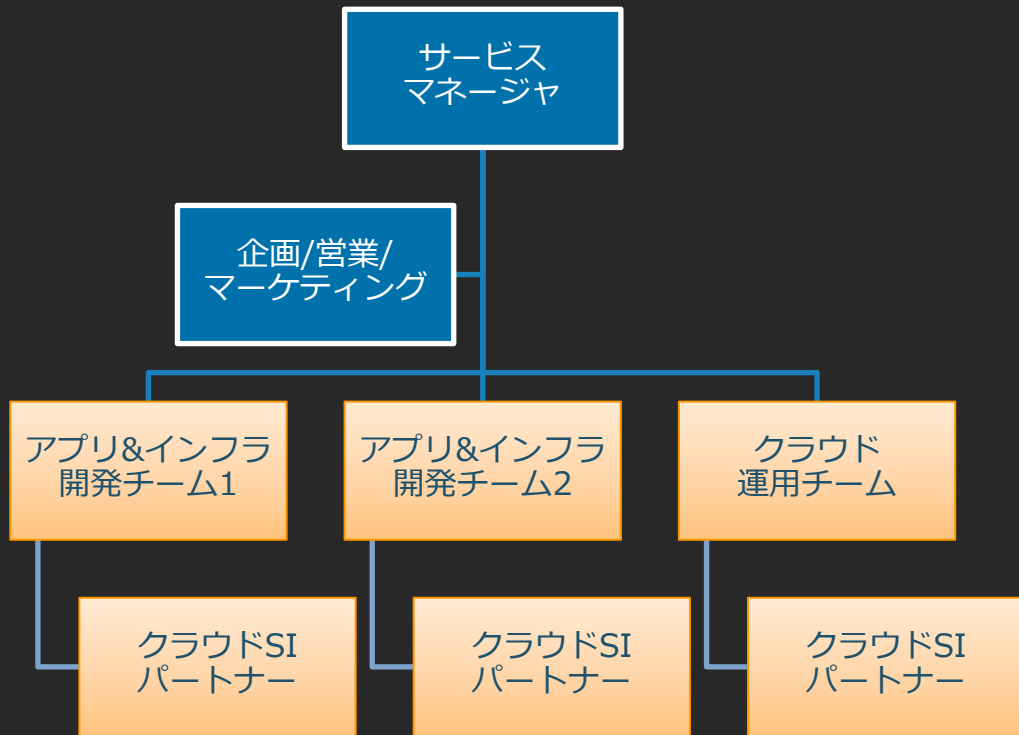


# クラウドを活用した開発体制



# クラウド上のアジャイル(DevOps)開発体制とスケジュール

## 開発・運用体制



## 開発スケジュール



クラウドによりインフラもアプリも  
ユーザからのフィードバックを得つつ  
開発することが可能に

# 運用における2つの選択肢

## 分散化とガードレール

- セルフサービス、実験、革新
- アジリティの向上
- DevOpsの実現
- 危険な操作の定義と対応
- 新メンバーの操作の監査

## ロックダウンと事前承認

- 完全な制御と実験余地の剥奪
- アジリティの低下
- 承認者数の増加
- 開発者が関与しない



# クラウドを活用したDevOpsのポイント

- ビジネス(アプリ開発)チームがアプリ/インフラ/運用まで担当
  - 利害関係の異なるチーム間でのコミュニケーションを削減
  - 自動化により Toil(手作業の運用などスピードを落とす業務)を減らす
- 共通基盤・運用チームはパターン化した環境の提供とガバナンスの確保をメインタスクとする
  - 開発チームのスキルレベルに合わせてセキュリティ設定を提供
  - 払い出したアカウント中で自由に操作可能とする(マルチアカウント)
- システムの一部をクラウド側に任せ、自動化することで迅速化、高機能化、低リスク、低コストを実現

# AWS Management Tools

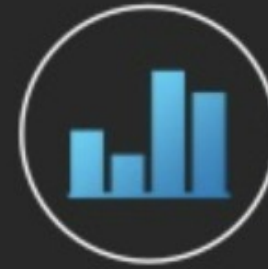
# AWS Management Tools



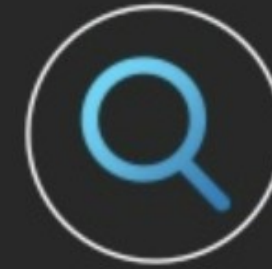
**Provisioning & entitlement**



**Configuration management**



**Monitoring**



**Operations and compliance management**

Integrated & interoperable

AWS CloudFormation  
AWS Service Catalog

AWS OpsWorks

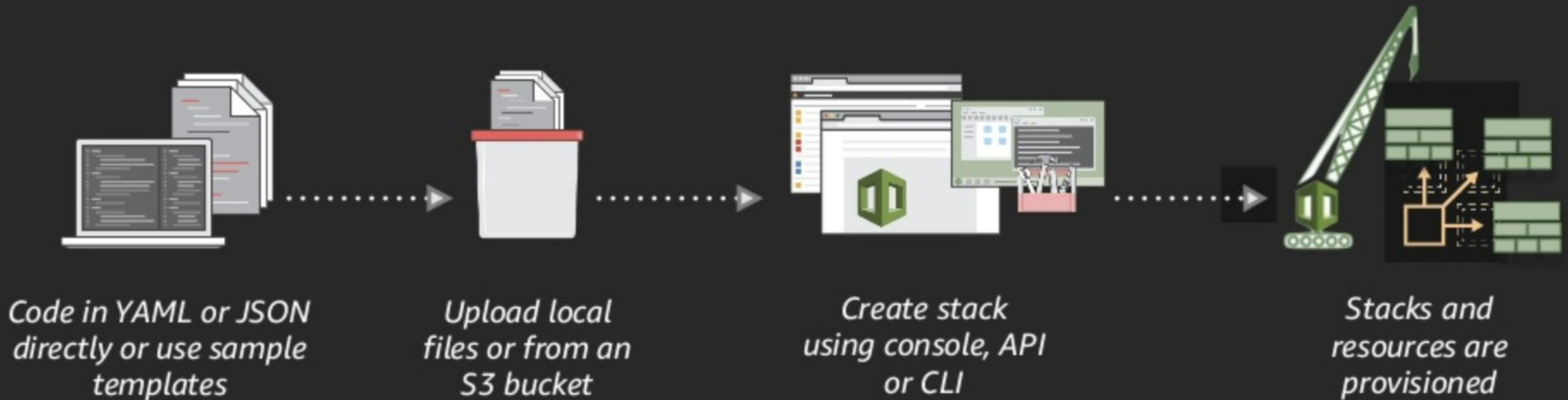
Amazon CloudWatch  
AWS X-Ray

AWS CloudTrail  
AWS Config  
AWS Systems Manager

# AWS CloudFormation



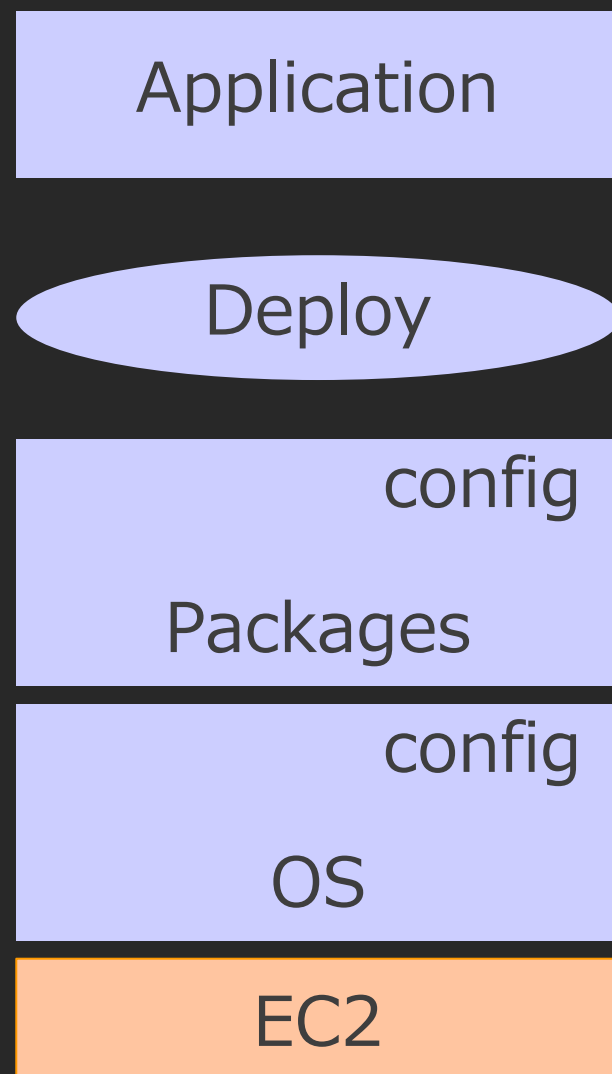
- 250以上のAWSリソースの作成を自動化
- リソースのアップデートを安全かつロールバック可能な形で実施
- サーバ、コンテナ、サーバレス、いずれのアプリ形態でも対応可能



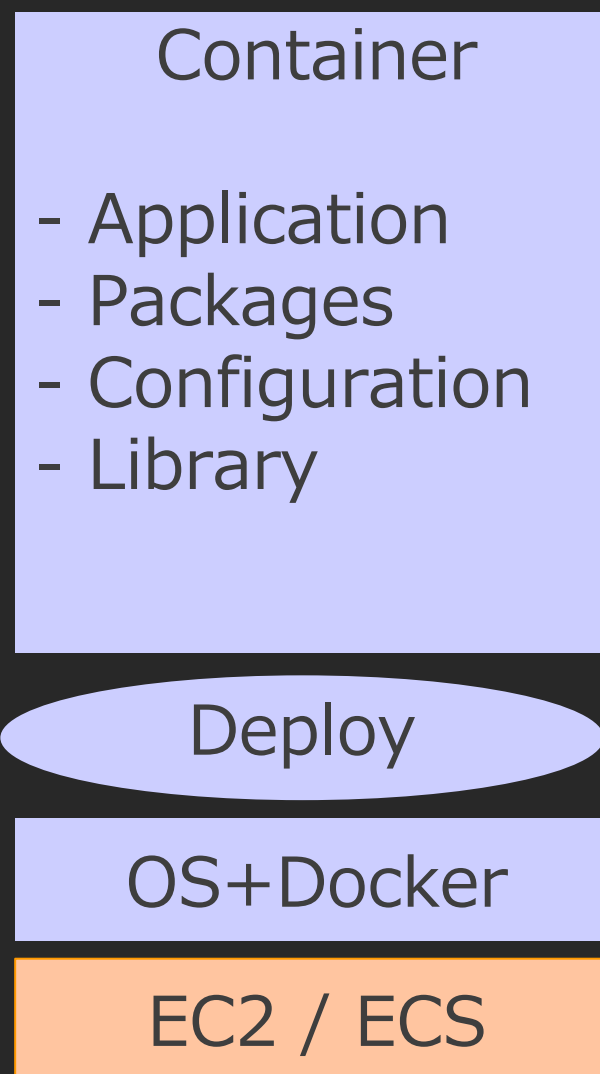
# 補足: アプリ形態によるデプロイ対象の違い

ユーザ管理

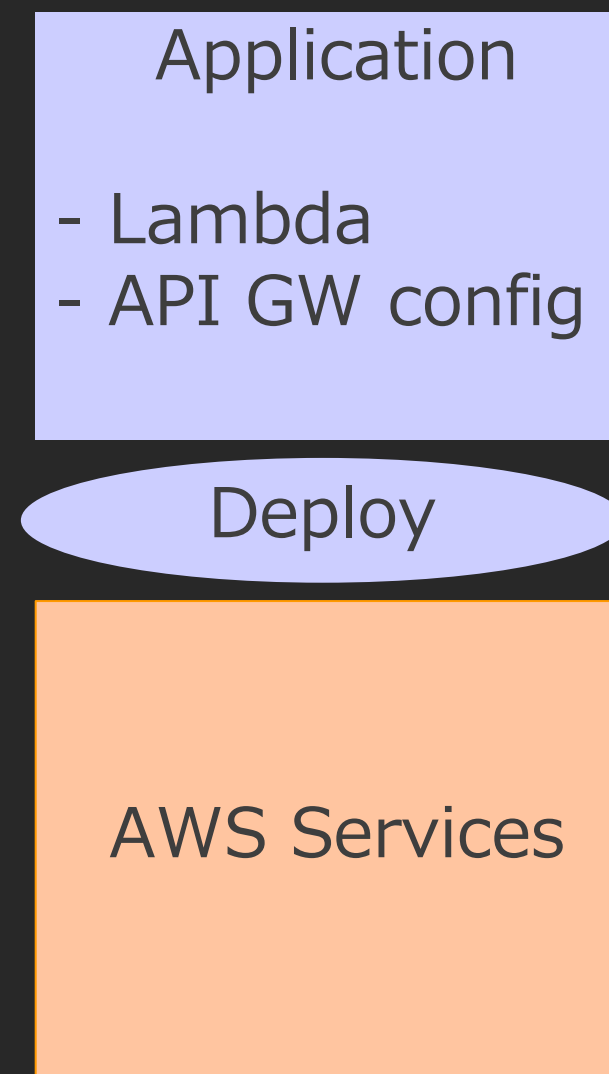
AWS管理



EC2

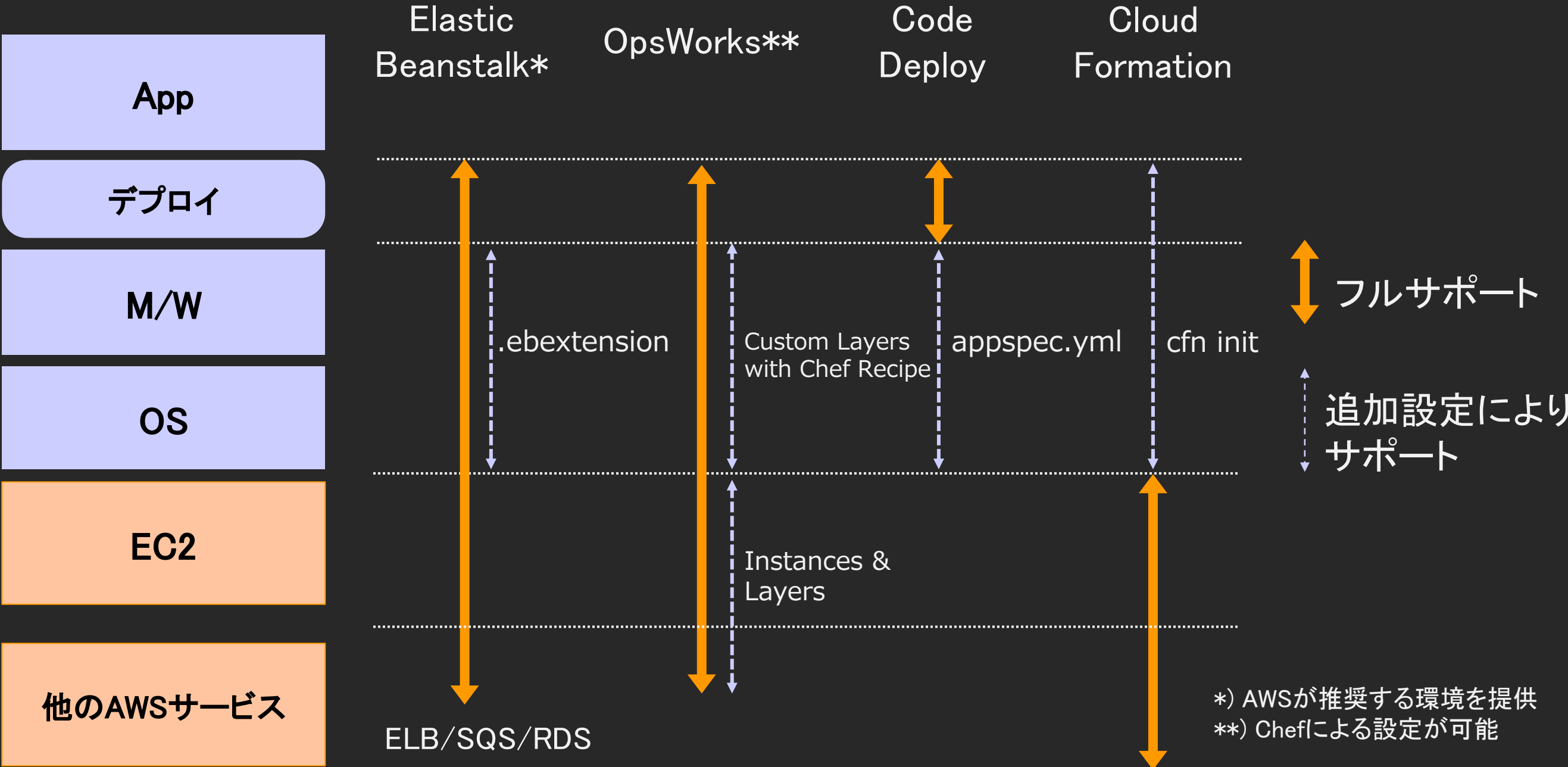


Container



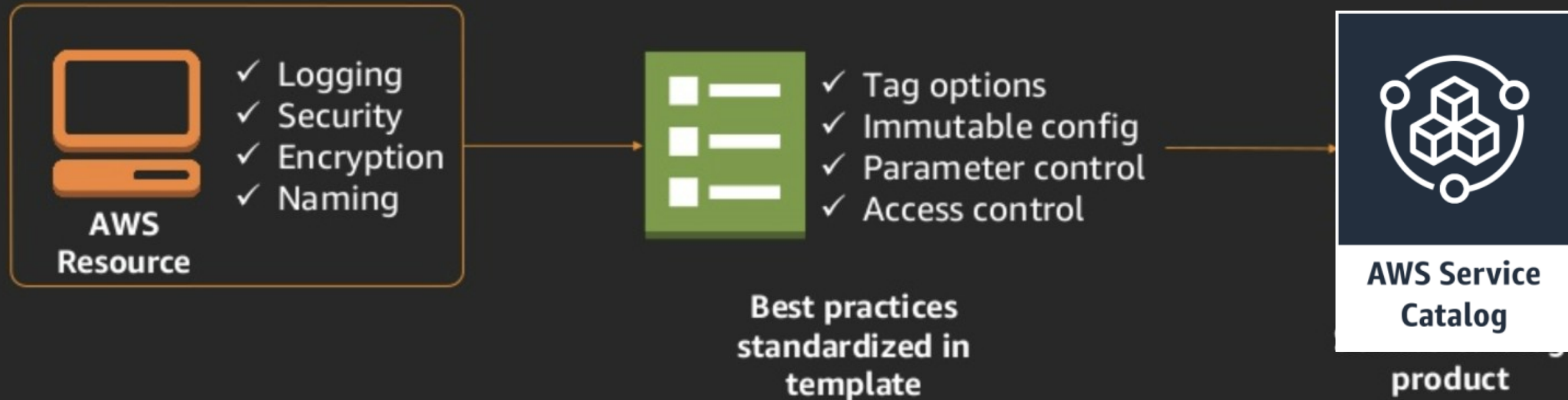
Serverless

# 補足:AWS プロビジョニングサービスのカバー範囲



\*) AWSが推奨する環境を提供  
 \*\*) Chefによる設定が可能

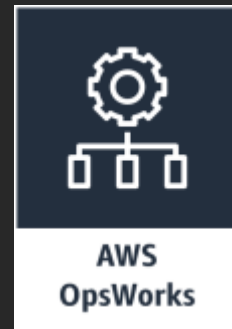
# AWS Service Catalog



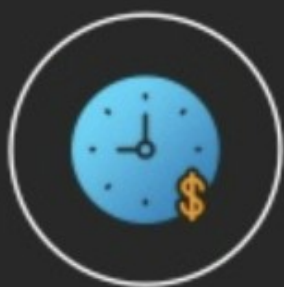
- 固定のベストプラクティステンプレートを作成 & 共有
- AWS サービスの限定されたアクセス制御 (SSMで一部操作のみ可能)
- エンドユーザは即時に使えるセルフサービス環境を利用可能



# AWS OpsWorks



- マネージドなコンフィグレーションサーバを提供
- Chef Automate と Puppet Enterprise に対応
- DSLを使用してコンフィグレーションを管理



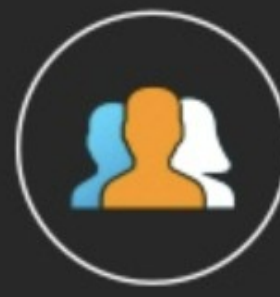
Pay-As-You-Go



Managed service



Backup and restore

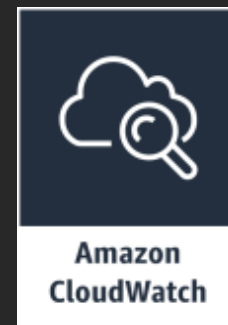


Support



Integration with AWS Services

# Amazon CloudWatch



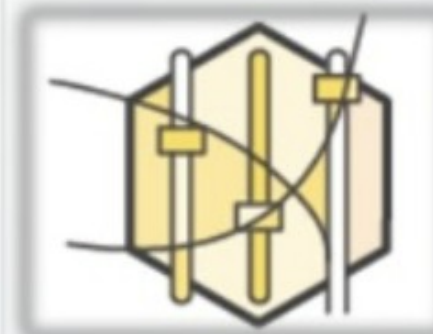
- AWS上のリソースのモニタリングと
- オンプレミス環境のモニタリングを実施
- メトリクスおよびログを管理



**Spot trends**



**Centralize monitoring**



**Troubleshoot**



**Monitor & store logs**



**Set alarms - events**

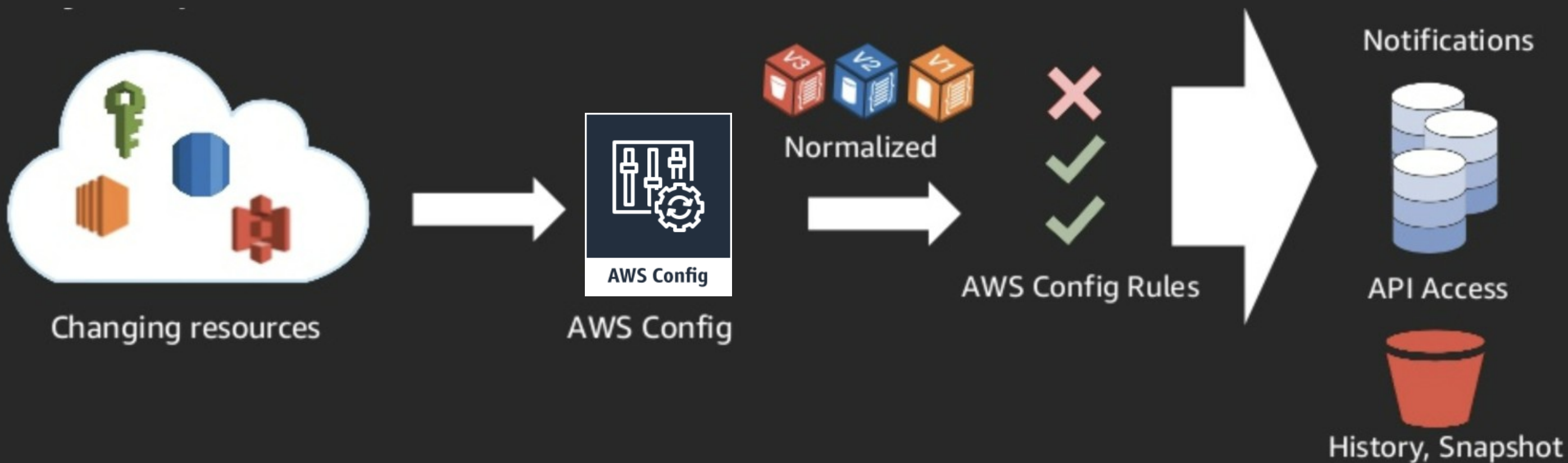


**Create dashboards**

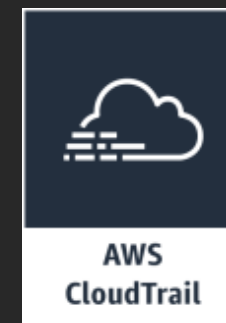
# AWS Config & AWS Config Rules



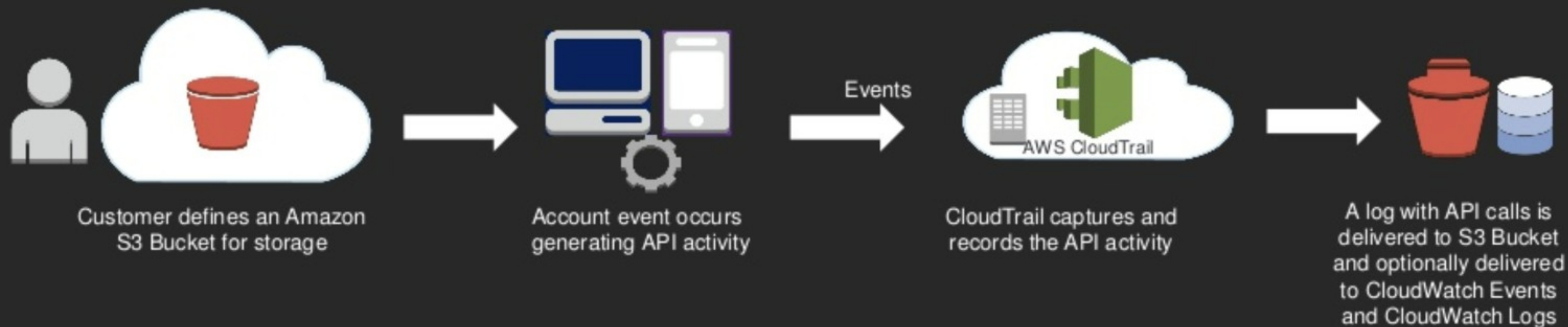
- 継続的記録、継続的アセスメントサービス
- AWSリソースの設定変更をトラッキング
- 自分で決めたポリシーにしがっていないとアラート発行



# AWS CloudTrail



- アカウントの活動(操作ログ)を自動的かつ集中して管理
- API usage event でセキュリティ監査と運用上のトラブルシューティングを実施



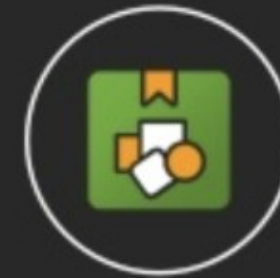
# AWS Systems Manager



Resource groups



Patch manager



State manager



Run command



Automation



Maintenance window



Inventory



Parameter store



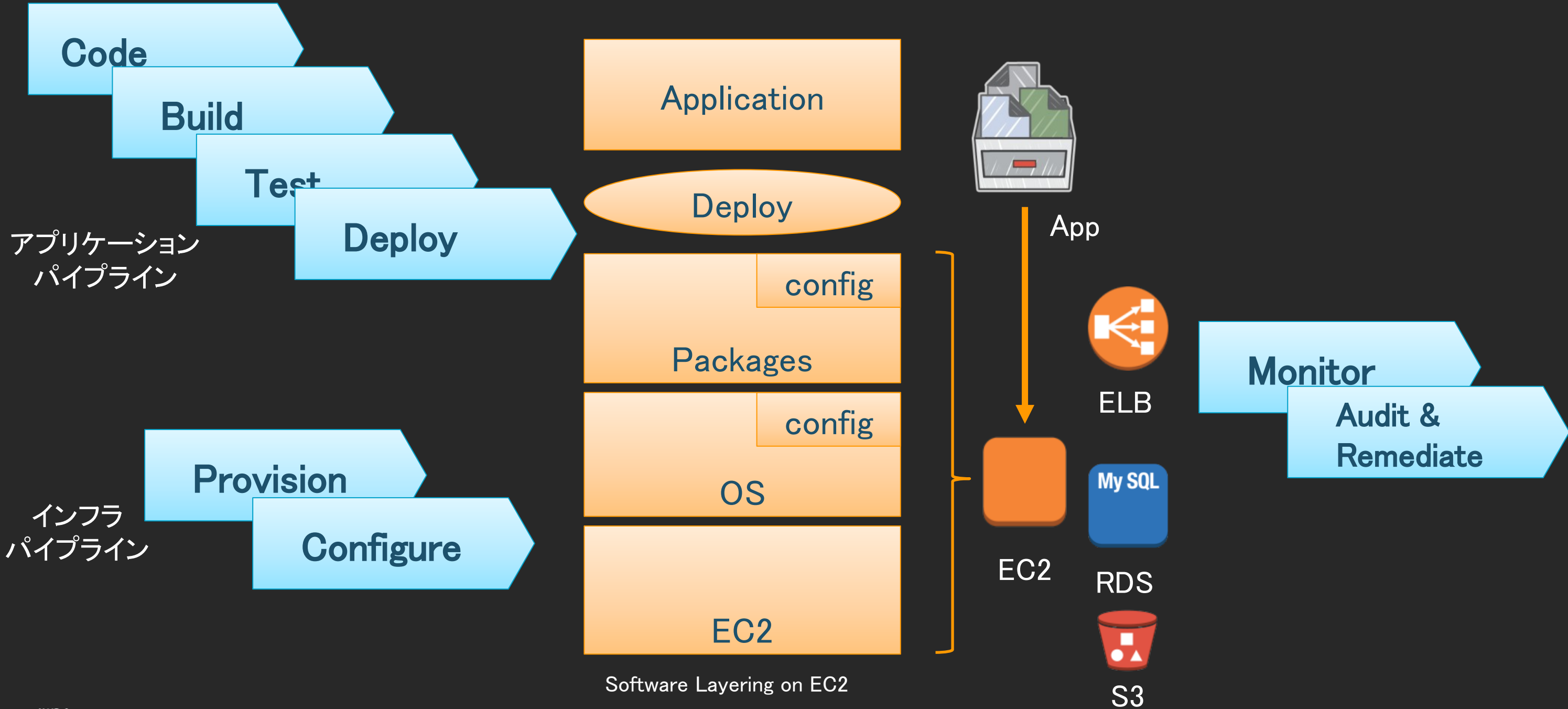
Session Manager



Distributor

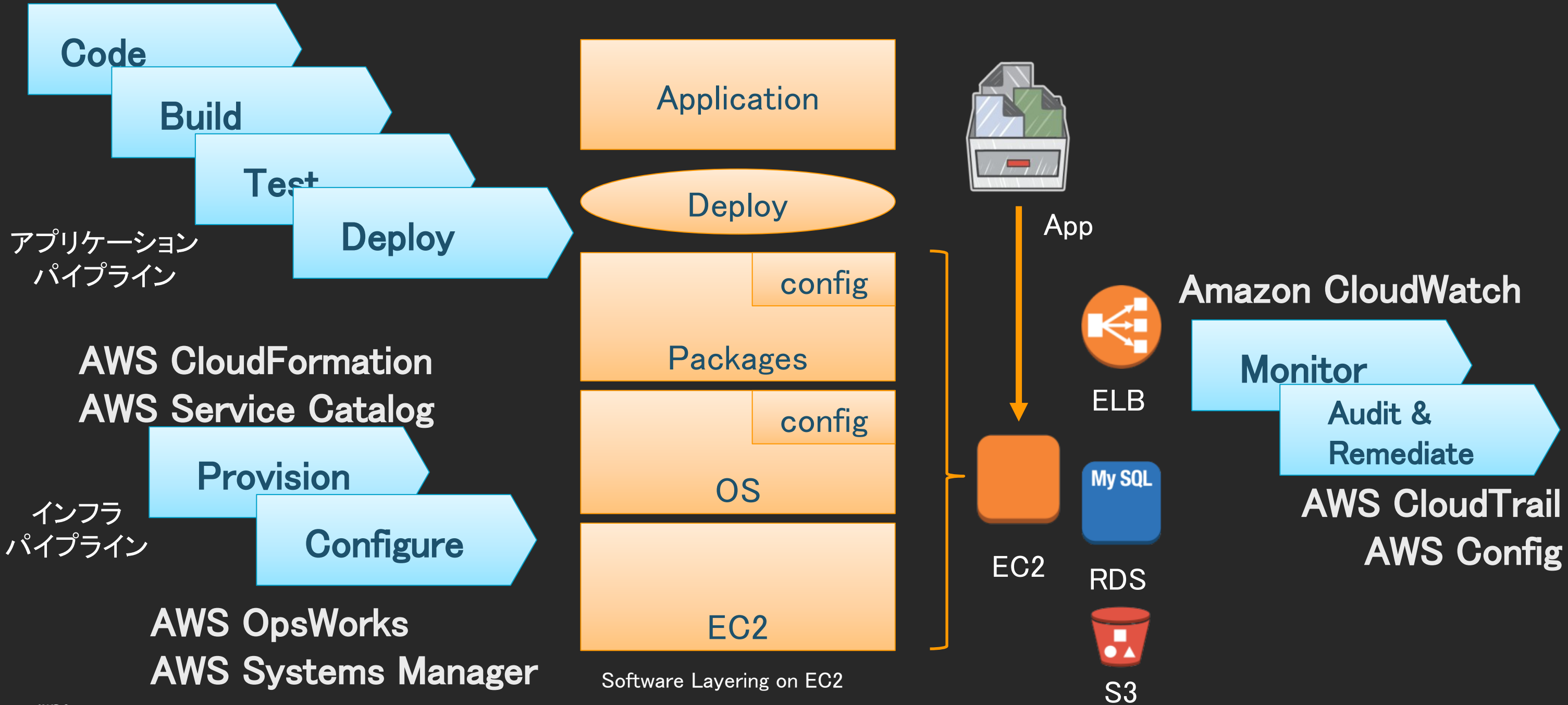


# Application & Infrastructure Pipelines



Software Layering on EC2

# Application & Infrastructure Pipelines





# 各サービスの使い所とアップデート

# Provisioning – CloudFormation (CFn)

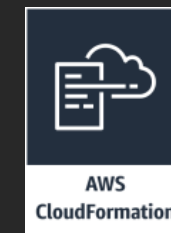
- |                          |                  |
|--------------------------|------------------|
| 1. テンプレート開発              | 各種ツールとプラグイン紹介    |
| 2. Cloud Development Kit | 一般の言語によるテンプレート開発 |
| 3. Drift Detection       | テンプレートとの差分検出     |
| 4. Macros                | テンプレート記述の拡充      |

2018/11/28 実施のBlackBelt オンラインセミナーもご覧ください

# CFn – テンプレート開発のサポート

- **cfn-lint**
  - Validate AWS CloudFormation yaml/json templates against the AWS CloudFormation spec and additional checks
- **cfn\_nag**
  - Look for patterns in AWS CloudFormation templates that may indicate insecure infrastructure.
- **Taskcat**
  - Catch problems that aren't obvious in a single template/stack
- **Cloud Development Kit**
  - Define cloud infrastructure in code and provision it through AWS CloudFormation
- **DSLs**
  - Troposphere/SparkleFormation/GoFormation

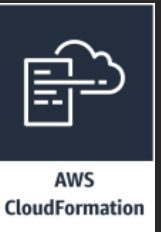
# CFn – テンプレート開発のサポート



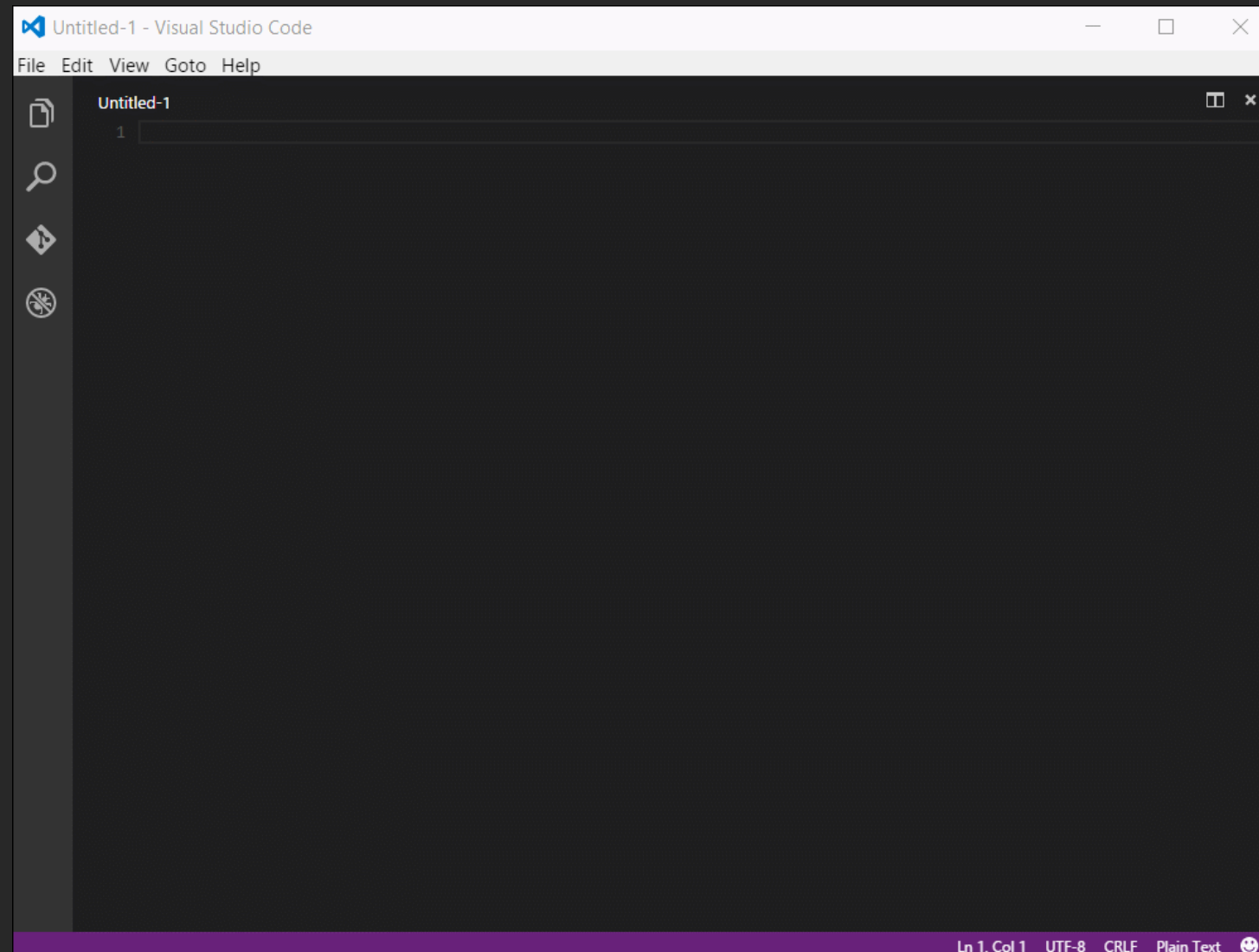
<https://github.com/aws-samples/aws-cloudformation-advanced-reinvent-2018>

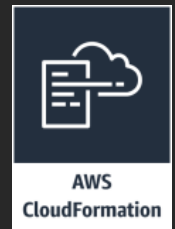
```
{ } User Settings | ! mynetwork-net.yaml ● | { } all-spec.json | ! 3-tier-app.cfn.yaml ●  
1 AWSTemplateFormatVersion: 2010-09-09  
2 |
```

# CloudFormation support for Visual Studio Code



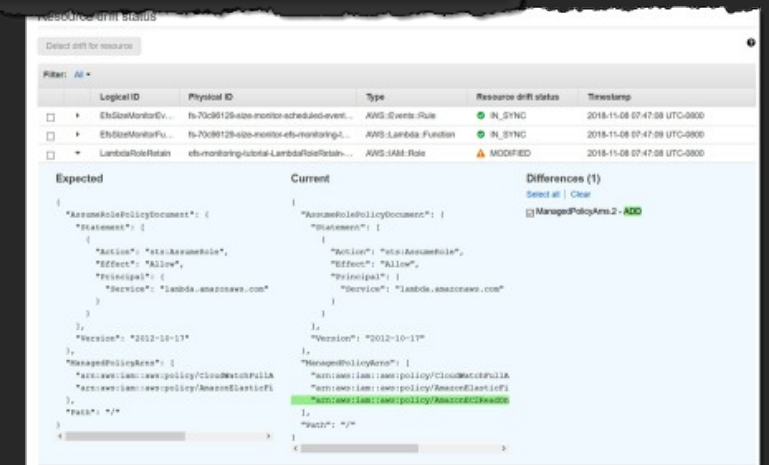
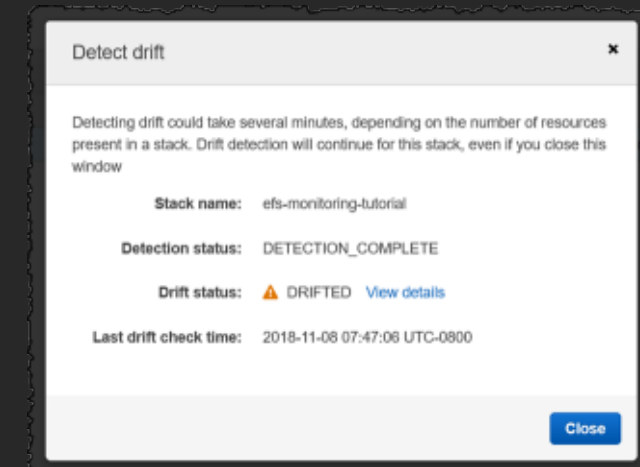
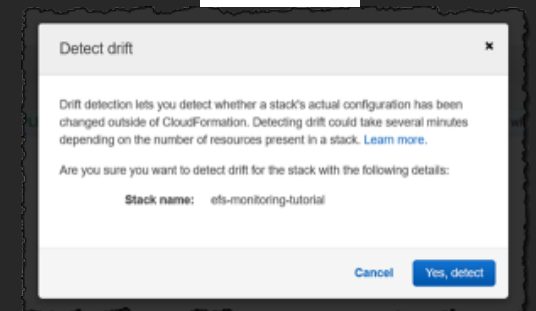
<https://github.com/aws-scripting-guy/cform-VSCode>





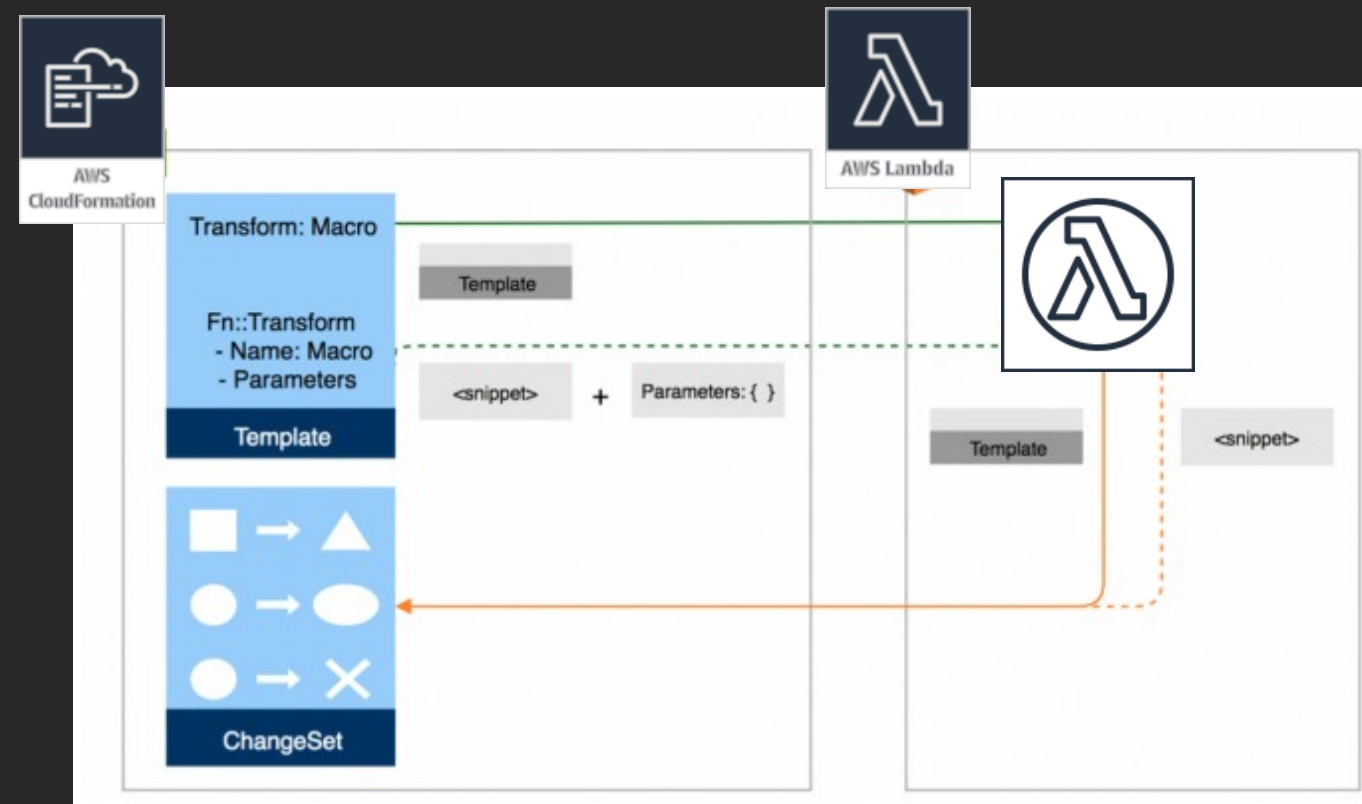
# CloudFormationでスタック差分の検出機能をリリース

- CloudFormationで作成したリソース状態と、現状との差分(Drift)を検出できるようになった
- スタックをデプロイした後に手動で実施した変更作業によって発生した差をチェックできる
- 現時点ではEC2, Auto Scaling, ECS, ELB, Lambda, RDSなどでサポート。詳細は下記  
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift-resource-list.html>
- AWS Configのマネージドルールもリリース。差分が発生したらすぐに検知できる



# Macros

- テンプレートの標準的な機能では実現できない処理を、Lambda関数を呼び出す事で実現
- 検索や置換などの単純な操作からテンプレート全体の変換まで独自処理が可能
- Lambdaでリソースの作成や削除を行わない事を推奨





# Macros

- Macros作成

```
AWSTemplateFormatVersion: "2010-09-09"  
Resources:  
  Macro:  
    Type: "AWS::CloudFormation::Macro"  
    Properties:  
      FunctionName: arn:aws:lambda:us-east-1:1234567:function:EchoFunction  
      Name: EchoMacro
```

Macrosリソースタイプ

Lambda関数 (arn) を指定

Macrosの名前

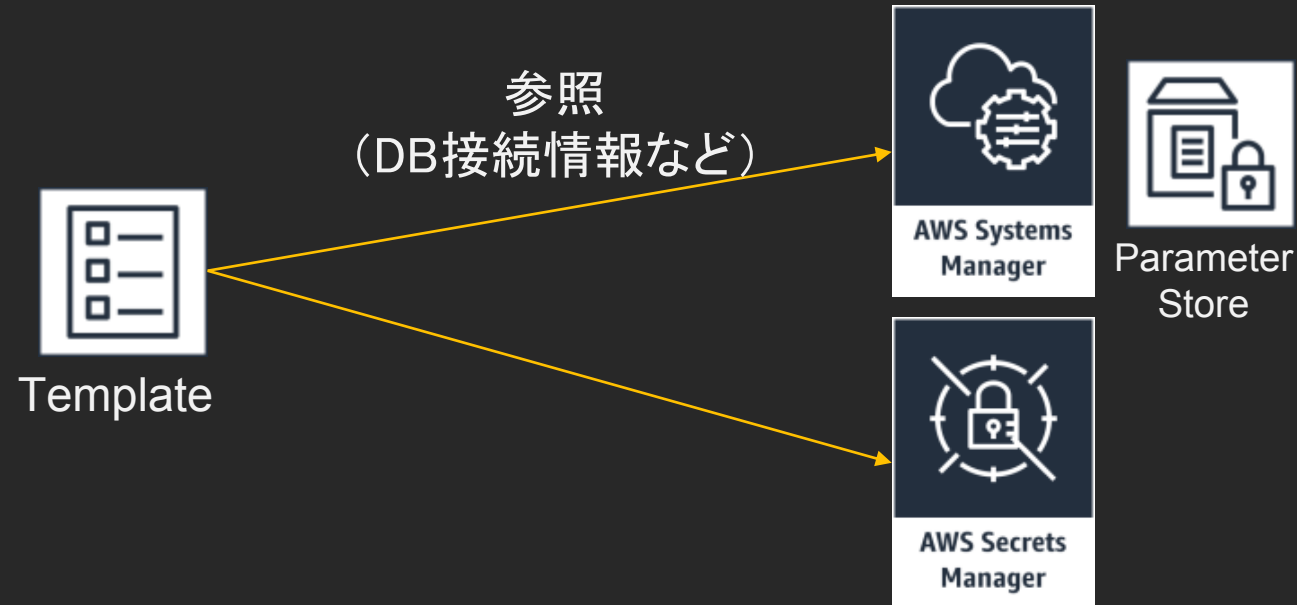
- Macros呼び出し

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: [EchoMacro, 'AWS::Serverless-2016-10-31']  
Resources:  
  FancyTable:  
    Type: AWS::Serverless::SimpleTable
```

呼び出すMacros

# Dynamic References

- 他のサービスに格納されたデータを動的に参照



サービス	テンプレート内の service-name	参照可能なデータ
AWS Systems Manager	ssm	パラメータストアに格納されているString/StringList（平文で保存されているデータ）
	ssm-secure	パラメータストアに格納されているSecureString（暗号化されて保存されているデータ）
AWS Secrets Manager	secretsmanager	保存されているすべてのシークレットまたは特定のシークレット

# Dynamic References

- 例) AWS Systems Managerに保存されたSecureStringを参照

AWS Systems Manager > パラメータストア

パラメータ 詳細

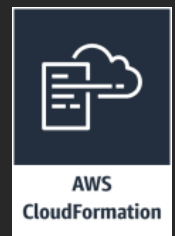
名前: begins-with: IAM

名前	種類	説明	キー ID	バージョン
IAMUserPassword-A	SecureString	-	alias/aws/ssm	1

AWS Systems Managerパラメータストアに  
IAMUserPassword-Aを保存

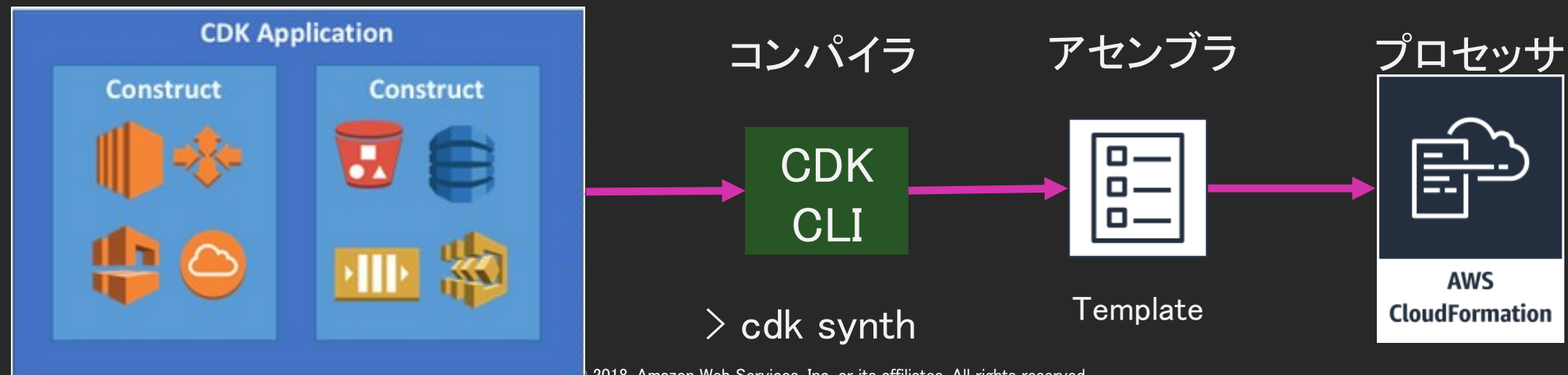
```
MyIAMUser:
  Type: AWS::IAM::User
  Properties:
    UserName: 'MyUserName'
    LoginProfile:
      Password: '{{resolve:ssm-secure:IAMUserPassword-A:1}}'
```

テンプレートから参照



# AWS CDK (AWS Cloud Development Kit)

- 開発者プレビュー
- ソフトウェア開発のフレームワーク
- インフラをJava等のコードで定義してAWS CloudFormationでプロビジョニング
- オブジェクト指向的に記述&ループ等が使用可能
- 推奨デフォルト値が設定済みのためコード量が少なく済む
- 現在、Java, JavaScript, TypeScript, .NET に対応



2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS CDK (AWS Cloud Development Kit)



Preview

- 例) S3バケット作成

```
const cdk = require('@aws-cdk/cdk');
const s3 = require('@aws-cdk/aws-s3');

class MyStack extends cdk.Stack {
  constructor(parent, id, props) {
    super(parent, id, props);

    new s3.Bucket(this, 'MyFirstBucket', {
      versioned: true
    });
  }
}
```

# Remediation – Systems Manager (SSM)

## New

1. Session Manager SSH/RDPやポート開放なくシェルアクセス
2. Distributer 独自パッケージの配布

## Updates

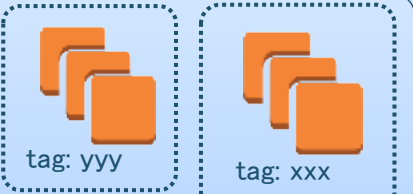
1. ResourceGroup リニューアルして77のリソースに対応、連携強化
2. Automation AWS API呼び出し,条件分岐,マルチアカウント対応
3. RunCommand Windows DSC対応
4. Maintenance Window スケジュール管理画面の強化
5. Inventory マルチアカウント対応とAthenaと連携した検索
6. Patch Manager Custom Patch Approval



# AWS Systems Manager によるサーバ運用

## マネージドインスタンス

AWS cloud

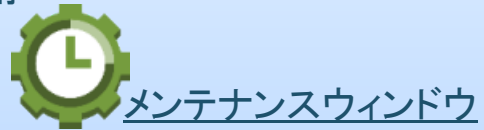


data center



## スケジューリング

定期運用



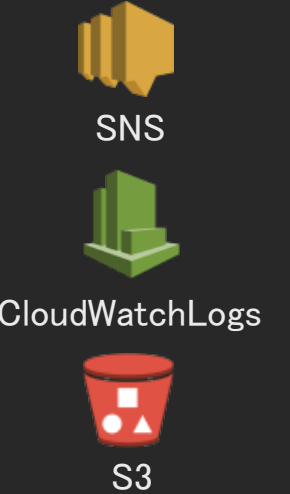
非定期・障害時運用



## オペレーション



操作記録・通知



IAM Role

SSM Agent

## サーバ運用

CloudWach Agent

## 構成検証・監査



## 構成情報・操作手順



記録・通知・対応・分析



パラメータストアの参照



© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

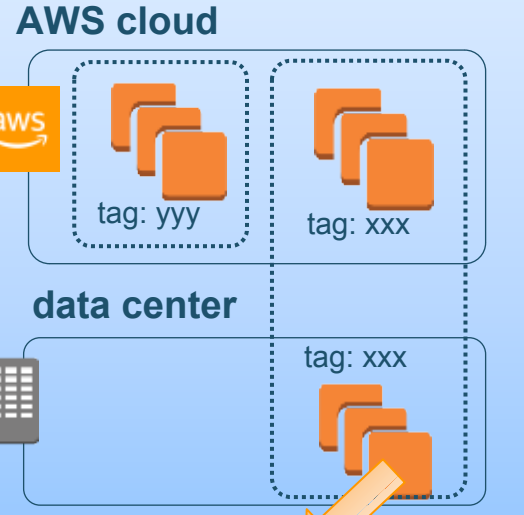
AWS re:Invent





# AWS Systems Manager によるサーバ運用

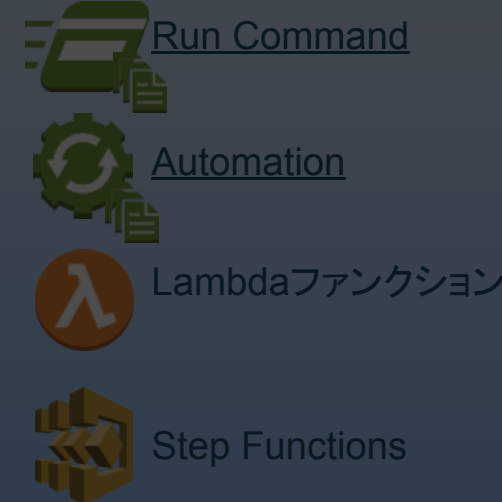
## マネージドインスタンス



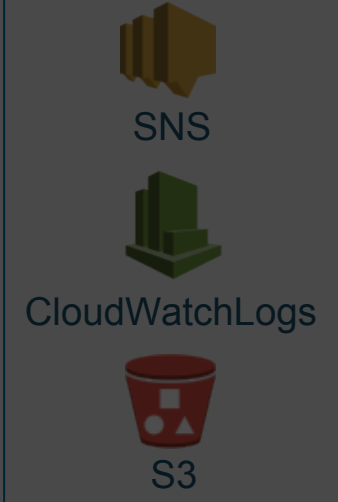
## スケジューリング



## オペレーション



## 操作記録・通知



IAM Role

SSM Agent

## サーバ運用

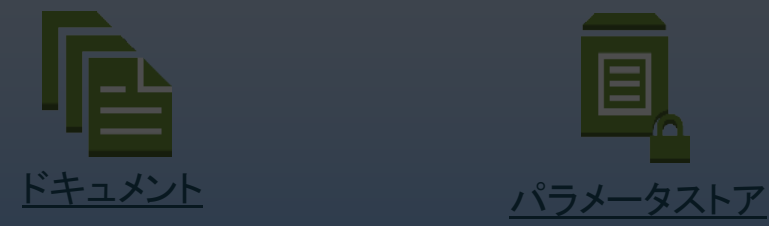
Inspector Agent

CloudWatch Agent

## 構成検証・監査



## 構成情報・操作手順



Inspector セキュリティ評価

CloudWatch メトリクス&ログ

## 記録・通知・対応・分析



## パラメータストアの参照





# AWS Systems Manager によるサーバ運用

## マネージドインスタンス

AWS cloud



## スケジューリング

定期運用



メンテナンスウィンドウ

非定期・障害時運用



マネジメントコンソール



Session Manager



## オペレーション



Run Command



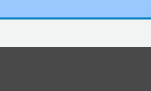
Automation



Distributer



Lambdaファンクション



Step Functions



操作記録・通知



SNS



CloudWatchLogs



S3

IAM Role



SSM Agent

## サーバ運用

Run Command

OS上の管理作業を事前定義済みのドキュメントを使用して安全に実施する  
SSHやRDPを使用せず多数のサーバを一括操作

Automation

AWS上の管理作業を事前定義済みのドキュメントを使用して安全に実施する  
パッチを適用してAMIを作成する、インスタンススペックを変更するなど

Distributer

独自のソフトウェアパッケージを定義し配布する

Session Manager

通信ポートを解放せずにサーバへのシェルアクセスを行う  
インタラクティブあるいは非定型の操作に使用

Athena&Quicksight

S3

AWS Config

CloudWatch Event

CFn

CLI

Lambda

ECS

# AWS Systems Manager によるサーバ運用

マネージドインスタンス

スケジューリング

オペレーション

操作記録・通知

AWS cloud

定期運用

Run Command

ステートマネージャ

あるべきOSの設定値を定義し維持する  
ポリシーに準拠したファイアウォール設定やアンチマルウェアツールの設定

インベントリ

オペレーティングシステム (OS)、アプリケーション、インスタンスのメタデータを収集する

Patch Manager

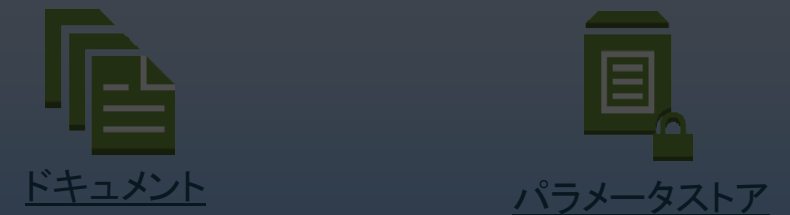
OSパッチの選択と適用を自動的に実施する

## サーバ運用

### 構成検証・監査



### 構成情報・操作手順



記録・通知・対応・分析



パラメータストアの参照



# AWS Systems Manager によるサーバ運用

マネージドインスタンス

スケジューリング

オペレーション

操作記録・通知

AWS cloud

定期運用

Run Command

SNS

ドキュメント

Systems Managerの各サービスの挙動を定義する設定ドキュメント  
コマンド、ポリシー、自動化の3種類がある

パラメータストア

設定値やパスワードなどを集中管理する階層型ストレージ  
アクセス制御や他のサービスからの参照が容易に行える

IAM Role

SSM Agent

サーバ運用

CloudWatch Agent

構成検証・監査

構成情報・操作手順

Inspector  
セキュリティ評価

CloudWatch  
メトリクス&ログ

ステートマネージャー

インベントリ

Patch Manager

ドキュメント

パラメータストア

記録・通知・対応・分析

Athena&Quicksight

S3

AWS Config

CloudWatch Event

パラメータストアの参照



CFn



CLI



Lambda



ECS

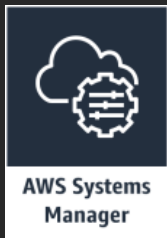






# AWS Systems Manager サービス群

サービス名	概要
メンテナンスウィンドウ	定期的に行う管理作業やサービス停止の実施時間帯を定義する
Run Command	OS上の管理作業を事前定義済みのドキュメントを使用して安全に実施する SSHやRDPを使用せず多数のサーバを一括操作
Automation	AWS上の管理作業を事前定義済みのドキュメントを使用して安全に実施する パッチを適用してAMIを作成する、インスタンススペックを変更するなど
Distributer	独自のソフトウェアパッケージを定義し配布する
Session Manager	通信ポートを解放せずにサーバへのシェルアクセスを行う インタラクティブあるいは非定型の操作に使用
ステートマネージャー	あるべきOSの設定値を定義し維持する ポリシーに準拠したファイアウォール設定やアンチマルウェアツールの設定
インベントリ	OS、アプリケーション、インスタンスのメタデータを収集する
Patch Manager	OSパッチの選択と適用を自動的に実施する
ドキュメント	Systems Managerの各サービスの挙動を定義する設定ドキュメント コマンド、ポリシー、自動化の3種類がある
パラメータストア	設定値やパスワードなどを集中管理する階層型ストレージ アクセス制御や他のサービスからの参照が容易に行える

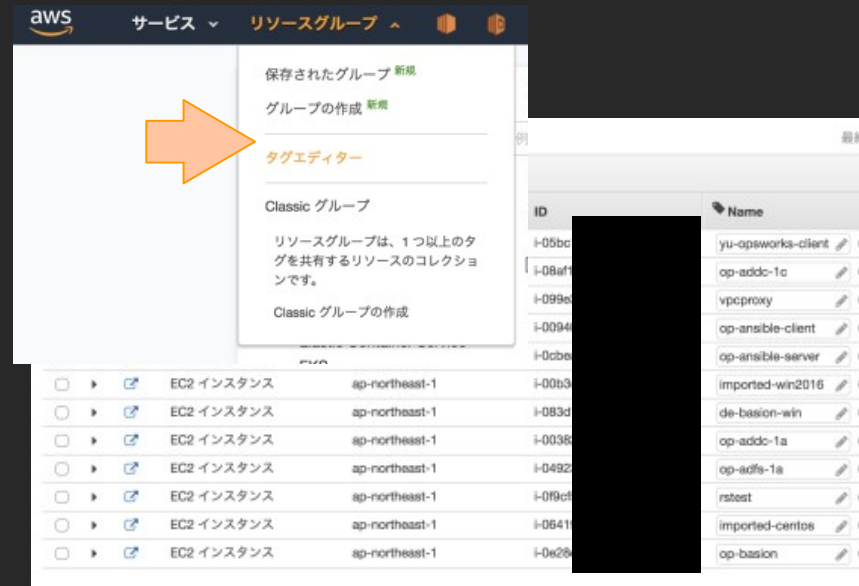


# Resource Group

- AWS上のリソースをグループ化して一括管理
- 77のリソースに対応
- タグエディタによる一括タグ付け

## Management Tools と統合

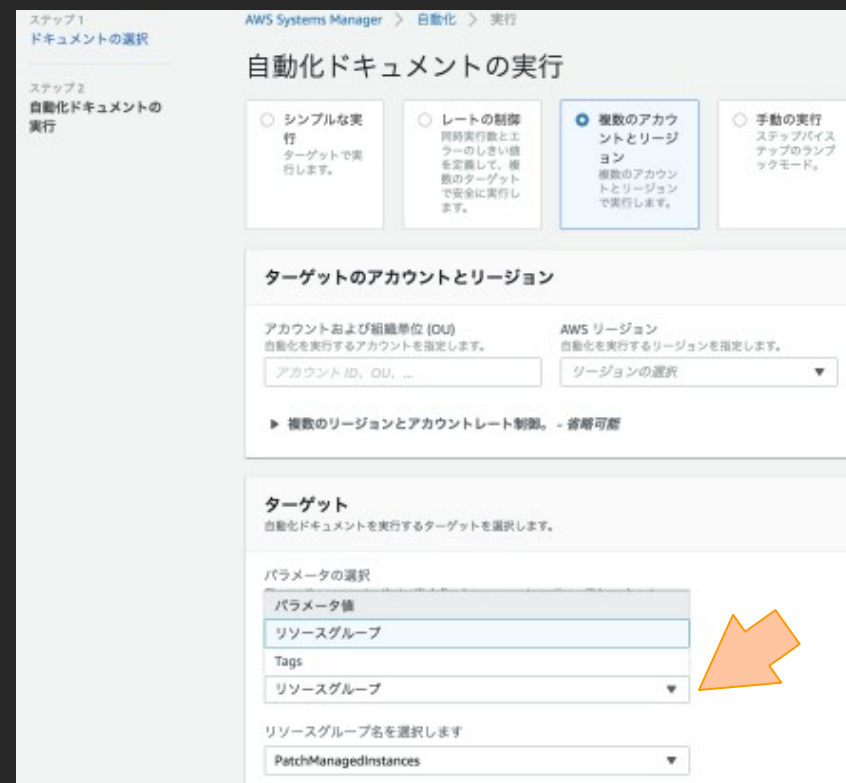
- Systems Manager
  - Automation, RunCommand, etc..
- CloudWatch Dashboard



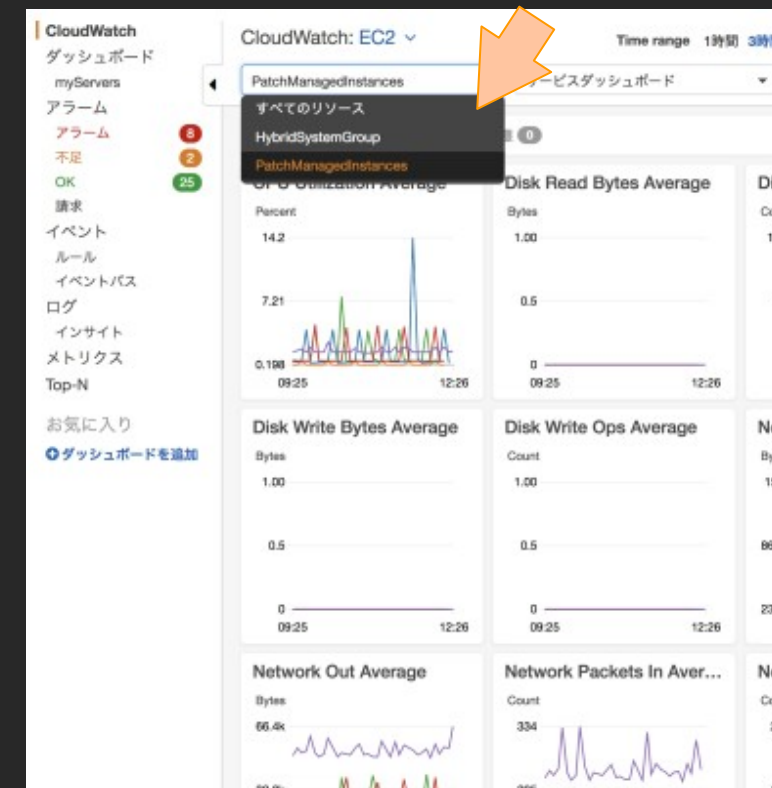
タグエディタ



Resource Group 作成

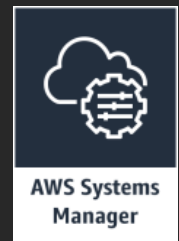


SSM Automationなど



CloudWatch





# AWS Systems Manager の Session Manager で マネジメントコンソールからOSへシェルアクセス

- 通信ポートを開放せずにサーバへのシェルアクセスが可能

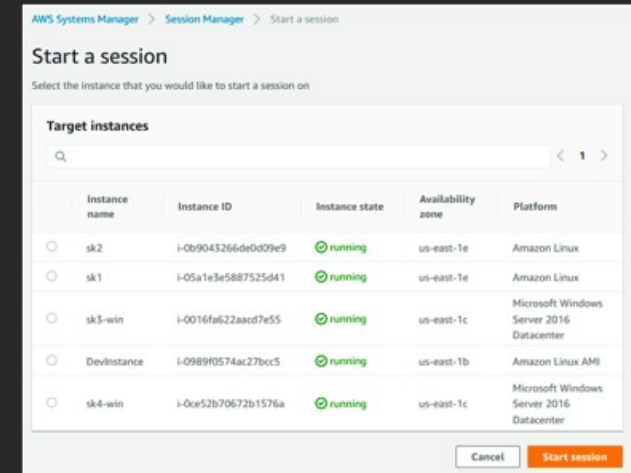
- マネジメントコンソールおよびCLIからアクセス
- Linux は bash、Windows は PowerShellが利用可能
- サーバのログイン情報(キーペアおよびID・パスワード)が不要

- その他の機能

- IAMユーザがアクセス可能なインスタンスをIAM Policyで指定可能
- 操作ログを CloudWatch Logs や S3 に保存。暗号化も可能。

- 前提 : SSM Agentの導入とエンドポイントへの通信が必要

- 対象サーバに最新の Systems Manager Agentが導入されていること
- 適切な権限(AmazonEC2RoleforSSMなど)を持つ IAM Role が EC2 に付与されていること
- インターネットへのアウトバウンドアクセスまたは、VPC内に作成したSystems Manager PrivateLink エンドポイントへの通信が行えること



```

Session ID: root-090c0eebbf6add0b0 Instance ID: i-0016fa622aad7e55
PS C:\Windows\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : eo2.Internal
    Link-local IPv6 Address . . . . . : fe80::fde7:9a39:1f04:6051%2
    IPv4 Address. . . . . : 172.31.28.77
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 172.31.16.1

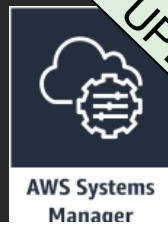
Tunnel adapter isatap.ec2.internal:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : eo2.Internal

Tunnel adapter Local Area Connection* 3:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2001:014137:9e74:10f9:1c0e:53e0:e3b2%2
    Link-local IPv6 Address . . . . . : fe80::10f9:1c0e:53e0:e3b2%2
    Default Gateway . . . . . :
  
```

# Automation で AWSの「操作手順」をYAML/JSONで記述

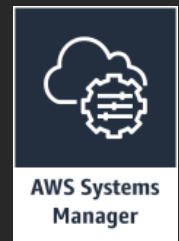


UPDATE

- 全てのAWS APIを呼び出せる executeAwsApi アクションが追加
  - 従来は事前定義済みのEC2に関する操作のみが指定可能
- Automation Documents (YAML or JSON)でAWS環境を操作するための「手続き」を記述できる
  - シェルやLambdaに比べ自由度は下がるが一定の書式やルールに従うことが可能
  - 簡単な処理フローも記述可能
  - AWS環境とサーバ上の操作を組み合わせたリカバリ処理などを定義できる

```
description: automate instance deployment
schemaVersion: '0.3'
mainSteps:
- name: getGoldenImageId
  action: aws:executeAwsApi
  inputs:
    Service: ssm
    Api: GetParameter
    Name: GoldenImageId
  outputs:
    - Name: Value
      Selector: "$.Parameter.Value"
      Type: String
- name: launch_ec2_instance
  action: aws:executeAwsApi
  inputs:
    Service: ec2
```





# AWS Systems Manager の Automation で パラメータによる条件分岐が可能に

EC2のOSタイプを取得して条件分岐する例

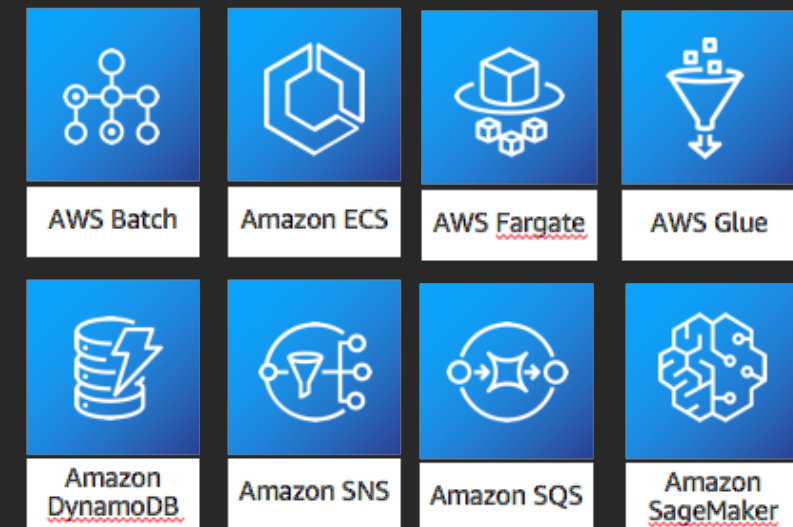
- Automation ドキュメント内で  
aws:branch を使うことで  
パラメータに応じた条件分岐が可能  
従来は前のステップの成功/失敗に  
応じた分岐のみが可能
- Systems Managerが利用可能な全てのリージョン

```
mainSteps:
- Name: GetInstance
  action: aws:executeAwsApi
  inputs:
    Service: ssm
    Api: DescribeInstanceInformation
  outputs:
- Name: myInstance
  Selector: "$.InstanceInformationList[0].InstanceId"
  Type: String
- Name: platform
  Selector: "$.InstanceInformationList[0].PlatformType"
  Type: String
- name: ChooseOSforCommand
  action: aws:branch
  inputs:
    Choices:
- NextStep: runPowerShellCommand
  Variable: "{{GetInstance.platform}}"
  StringEquals: Windows
- NextStep: runShellCommand
  Variable: "{{GetInstance.platform}}"
  StringEquals: Linux
  Default:
    Sleep
```



# 様々なサービスをプログラミングレスでつなぎ合わせる AWS Step Functions API Connectorsを発表

- Step Functionsのステートマシンから他のAWSサービス群に対して直接操作できるようになり、Lambdaを介在させる必要がなくなった
  - ✓ DynamoDB: 既存のテーブルからitemの取り出し、新規itemの追加
  - ✓ AWS Batch: バッチジョブの開始と完了待機
  - ✓ Amazon ECS/Fargate: ECSまたはFargateのタスクを実行する
  - ✓ Amazon SNS: SNSトピックにメッセージをパブリッシュする
  - ✓ Amazon SQS: キューにメッセージをプッシュする
  - ✓ AWS Glue: ジョブを開始する
  - ✓ Amazon SageMaker: 学習ジョブ、変換ジョブを開始する

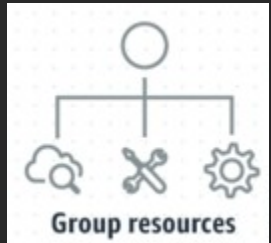


# AWS Systems Manager パッケージ管理機能 (Distributer) の追加

- 独自のソフトウェアパッケージを作成してサーバへ配布
  - ✓ ソフトウェアと install / uninstall スクリプトを zip 化してパッケージ登録
  - ✓ 配布対象のサーバ群を定義しアクセスコントロール
  - ✓ RunCommand (1回) や StateManager (定期) を使って配布



AWS Systems  
Manager



Group resources



Visualize data



Take action

# AWS Systems Manager Inventory および Automation のマルチアカウント対応



UPDATE

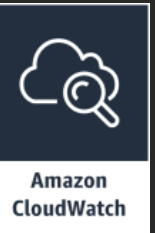
- Inventory
  - OS上のソフトウェア構成を収集し Athenaでアカウントやリージョンをまたいで検索
- Automation
  - キュメントに定義した一連のAPI操作をアカウントやリージョンをまたいだAWSリソースに対して実行

The screenshot shows the AWS Systems Manager Inventory console. At the top, there are navigation tabs for 'ダッシュボード', '詳細ビュー', and '設定'. A blue information box states: 'この機能では、AWS Athena、AWS Glue、リソースデータの同期を使用してインベントリデータを表示します。この機能を使用するには、リソースデータの同期を選択する必要があります。料金が適用される場合があります。' Below this, there are buttons for 'リソースデータの同期', 'インベントリデータの表示', and 'リソースデータの同期の作成'. A search bar is labeled '同期名によるフィルター'. A table displays synchronization details:

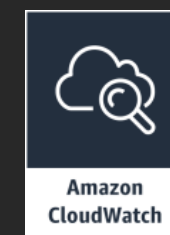
同期名	同期日 (UTC)	前回のステータス	最終の同期 (UTC)	最後に成功した同期 (UTC)
ssm-resource	Wed, 11 Jul 2018 07:10:31 GMT	成功	Sun, 02 Dec 2018 22:46:05 GMT	Sun, 02 Dec 2018 22:46:05 GMT

Below the table, there is a section for 'インベントリタイプ' with a dropdown menu set to 'AWS:Application'. Underneath, there are buttons for 'CSVへエクスポート', 'Query History', and 'Run Advanced Queries'. A search bar is present, and a pagination bar shows '1 2 3 4 5 6 7 8 ...'. At the bottom, a table header includes 'Package ID', 'Account ID', 'Region', and 'Publisher'.

# Monitoring – CloudWatch

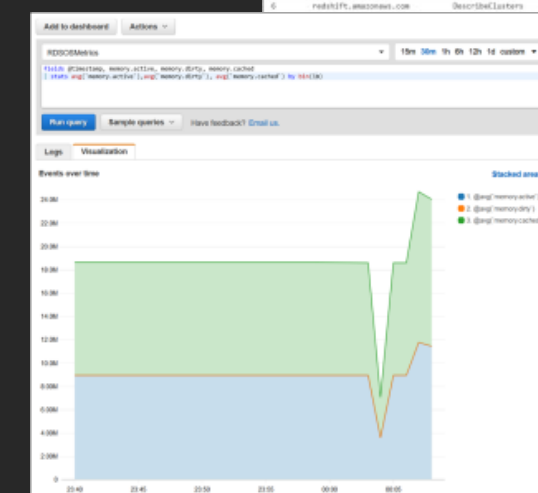
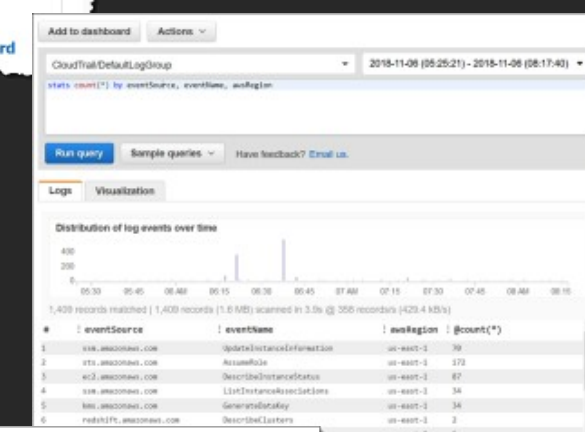
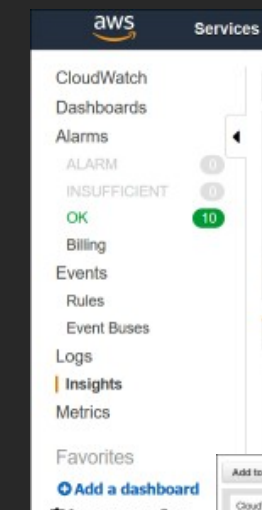


- |                               |                   |
|-------------------------------|-------------------|
| 1. CloudWatch Logs Insight    | 収集ログをクエリでリアルタイム分析 |
| 2. Automated Dashboard        | 集約ダッシュボードの自動生成    |
| 3. Metric Math Alarm          | 計算値に基づくアラームの発行    |
| 4. CloudWatch Snapshot Graphs | グラフをマネコン以外で利用     |

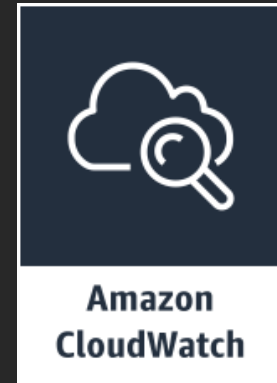


# インタラクティブなログ分析を実現する Amazon CloudWatch Logs Insightsを発表

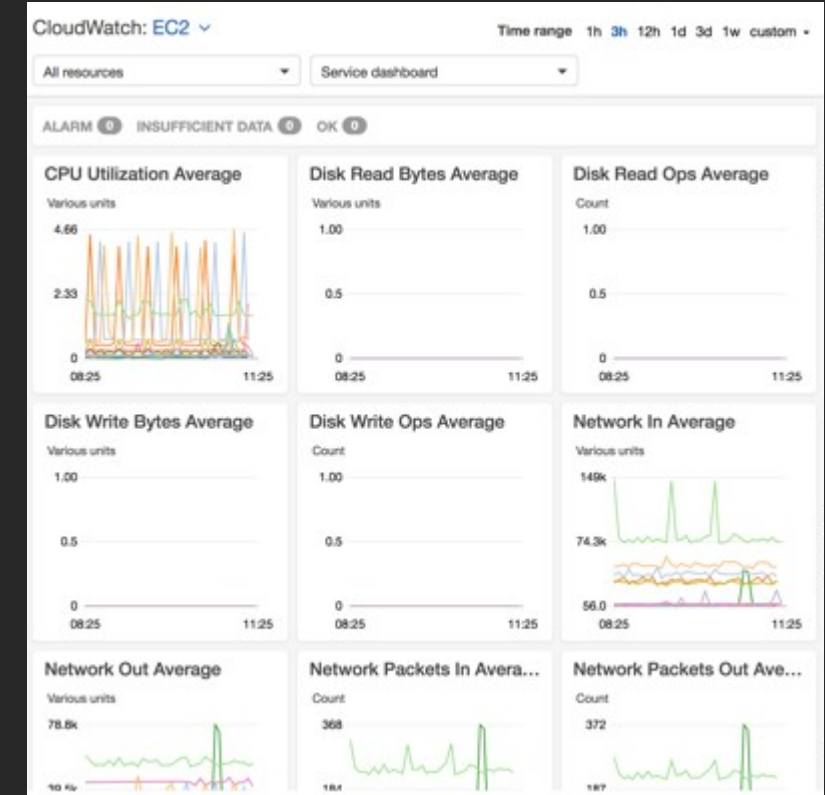
- AWSが生成する各種ログや、様々なサーバアプリケーションのログの分析や可視化を実現するフルマネージドサービス
- CloudWatch Logsで収集したログに対して、クエリ言語を利用してソースや時間でのフィルタリング・ビジュアライズを実行できる
- クエリの結果をDashboardに追加することも
- 東京を含む各リージョンで利用可能。クエリによりスキャンされたログデータ量に応じた課金があり、\$0.0076/GB(東京)となる



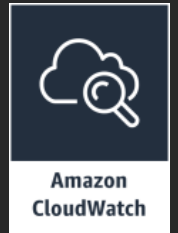
# CloudWatch が Automatic Dashboard と 計算値に基づくアラームに対応



- AWSが推奨するベストプラクティスに基づいた集約ダッシュボードを自動生成
  - ✓ 各主要サービスごとあるいはサービスをまたいだダッシュボード
  - ✓ リソースグループを使用してシステムごとにフィルタリング
- 複数のメトリックの計算結果に基づきアラーム発行
  - ✓ 計算値メトリックのグラフはすでに利用可能だったがアラーム発行も計算値に基づいて可能に
  - ✓ +, -, /, \* や Sum, Average, Min, Max などが利用可能

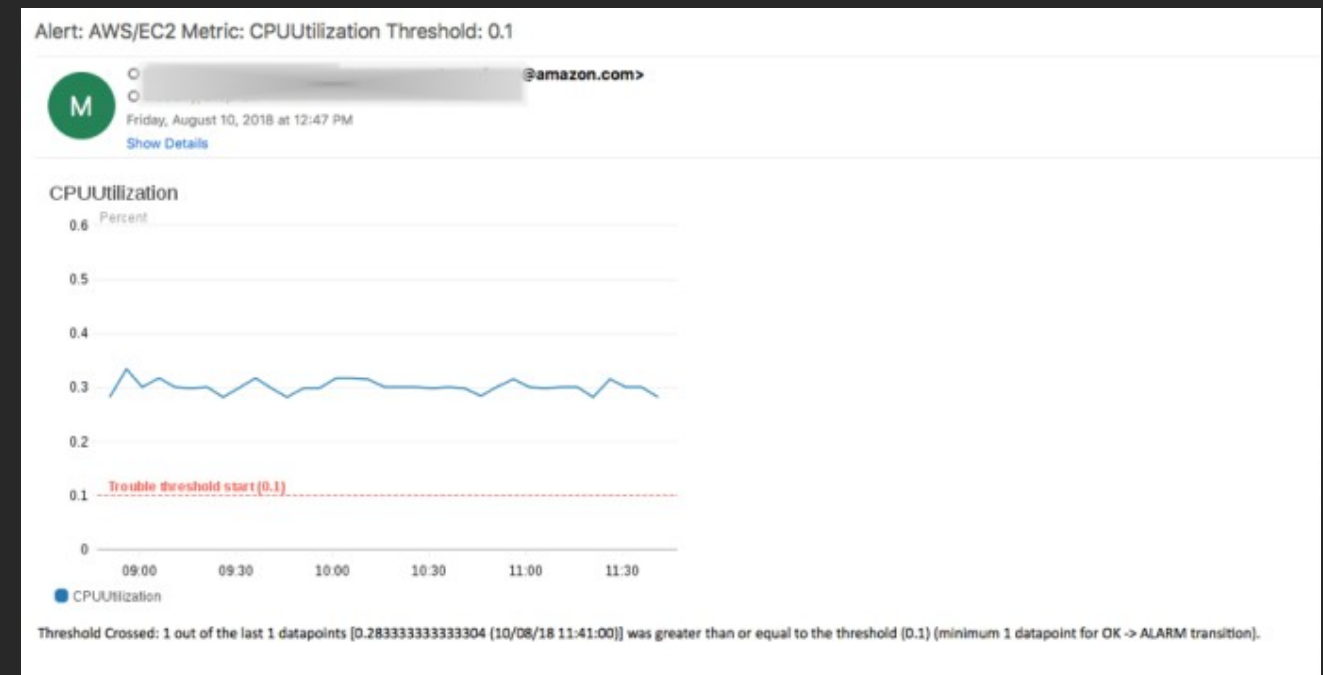
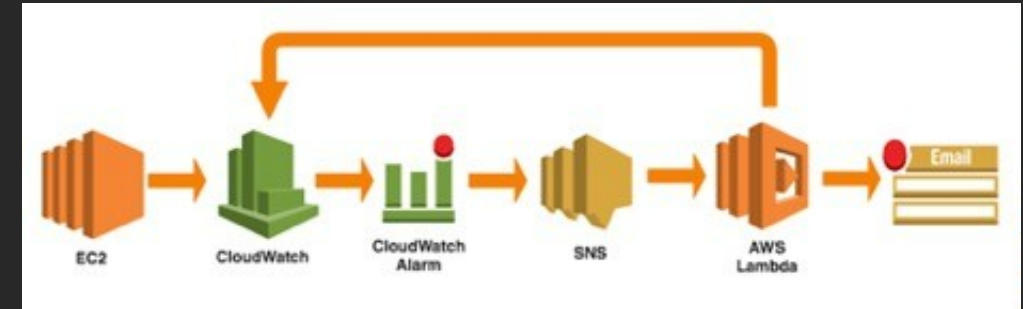






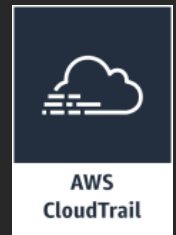
# CloudWatch のグラフを外部から取得可能に

- CloudWatchで作成したグラフの画像イメージをAPIで取得可能
- マネジメントコンソール外でグラフを参照可能
  - CloudWatch Alarmと組み合わせてSlackやメールに異常時のグラフを添付する
  - プロジェクトのポータルサイトにCloudWatchのグラフを表示する





# Audit – CloudTrail & AWS Config & Config Rules



1. Config Rules – Detect Drift for CFnCFnテンプレートとの差異を検出
2. AWS Config Multi-Account Aggregation マルチアカウント情報集約

# マルチアカウント関連アップデート

# VPCとアカウントの分離レベルの違い

## VPCによる分離

- VPC間通信はデフォルトNG\*1
- APIアクセスは共有(IAMで制御)

## アカウントによる分離

- VPC間通信はデフォルトNG\*1
- API相互アクセスはデフォルトNG\*2

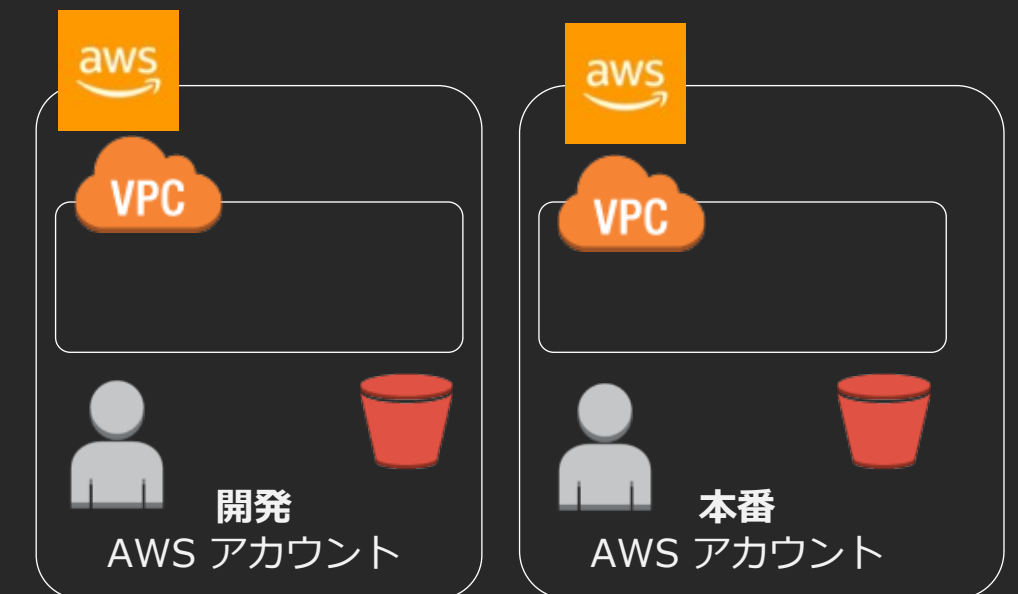
\*1 - VPC Peeringを結ぶことで相互に通信可能

\*2 - IAMのクロスアカウントアクセスで相互にアクセス可能

## VPC による分離



## アカウントによる分離



# Private Link によるプライベートネットワーク経由APIアクセス

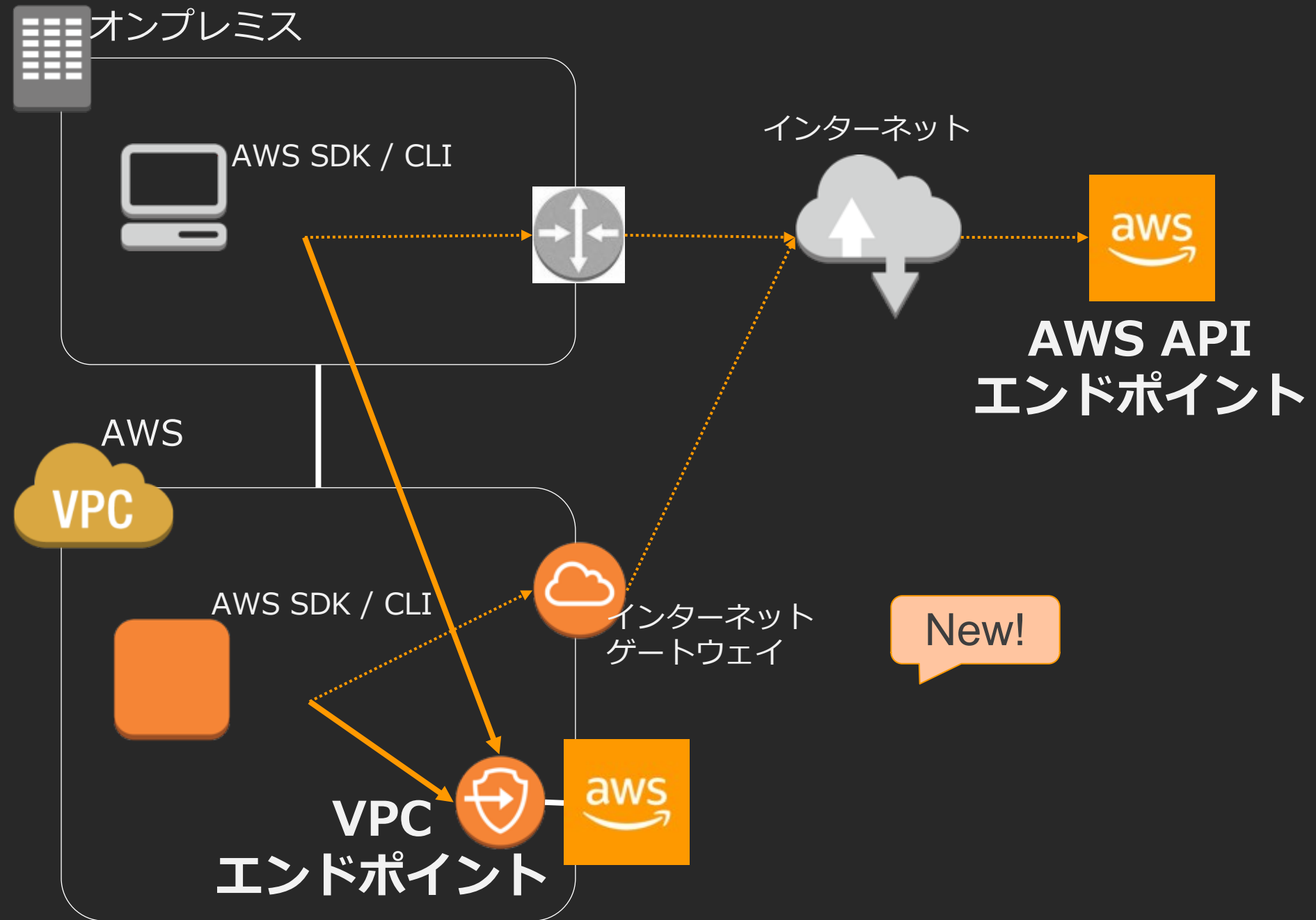
通常のAPIアクセスはインターネット経由

VPCエンドポイントでAPIに対してプライベートアクセスが可能

専用線 (Direct Connect)

VPN

New!



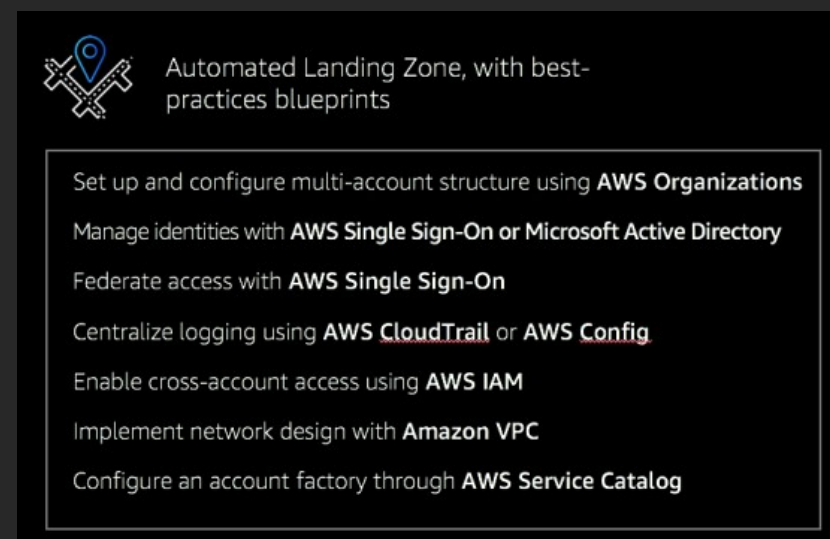
# マルチアカウント関連アップデート

アカウントレベルの払い出し管理がより容易に

1. アカウント作成 – Control Tower
2. リソース共有 – Resource Access Manager, Shared VPC
3. アクセス可能 – Transit GW, EFS
4. Provisioning – CFn StackSets
5. Monitoring – CloudWatch Events Bus
6. Audit – Security Hub, AWS Config, SSM Inventory
7. Remediation – SSM Automation

# マルチアカウント環境におけるガバナンスを強化する AWS Control Towerを発表

- 複数のアカウントに組織として必要とするセキュリティやコンプライアンスを確保し、ベストプラクティスに沿ったシステムを展開することを容易に
- システムがルールに沿っているかをハイレベルな視点でチェックし、必須または推奨のレベルで通知する機能を備えており、準拠状況をモニタできる
- 現状を把握するためのダッシュボードが備わっており、準拠すべきルールにどの程度従えているかを視覚的に把握する機能も
- Control Towerは無料で利用できる



# クロスアカウントでのリソース共有が可能に AWS Resource Access Manager を発表

- アカウント間で同じリソースを共有して利用することが可能に
- 現時点の対象リソース
  - VPC サブネット
  - R53 リゾルバルール
  - License Manager - License Configuration
- 自分のリソースを Organization、Organization Unit (OU)、AWSアカウントで共有（再共有はNG）
- すべての商用リージョンで利用可能

The screenshot displays the AWS Resource Access Manager console. On the left, a navigation pane shows 'Resource Access Manager' with sections for 'Shared by me' (Resource shares, Shared resources, Principals) and 'Shared with me' (Resource shares, Shared resources, Principals, Settings). The main content area shows the details for a 'BaseSetShare (rs-ccb3b87e-91ba-59ca-ea8b-a6cc4b3473b2)'. It includes a 'Summary' table and a 'Shared resources (2)' table.

Name	Owner	Created on	Status
BaseSetShare	340*****	2018/12/02	Active

ID	ARN	Allow external principals
rs-ccb3b87e-91ba-59ca-ea8b-a6cc4b3473b2	arn:aws:ram:ap-northeast-1:340*****:resource-share/ccb3b87e-91ba-59ca-ea8b-a6cc4b3473b2	No

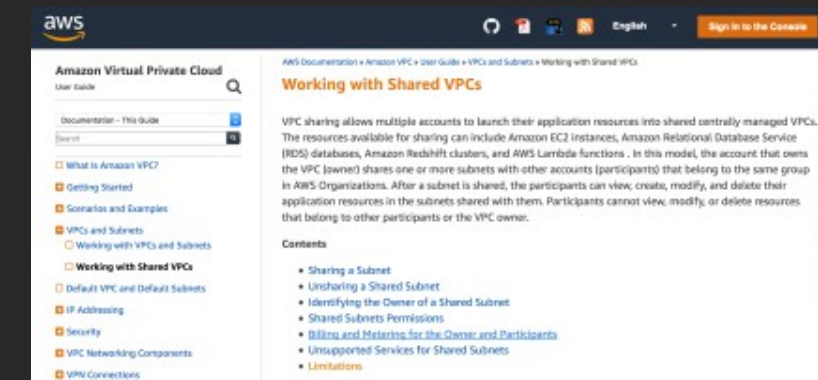
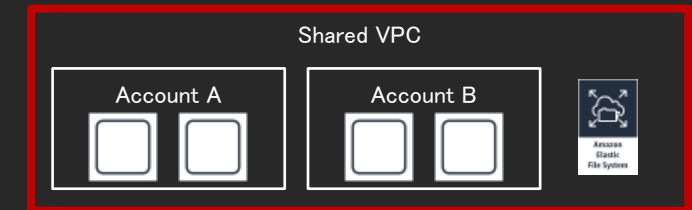
  

Resource ID	Resource type	Status
rslvr-rr-d75*****b	route53resolver:ResolverRule	Associated
subnet-00c*****7d	ec2:Subnet	Associated



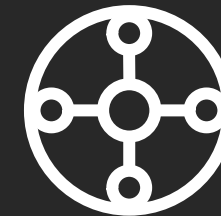
# 複数のAWSアカウントで1つのVPCを共有できる Shared VPCを発表

- これまでは複数のVPCとピアリングによって構成する必要があったが、単一のVPCで実現可能になった
- 設計上1VPCに集約することが必要なケースでも、複数のチームに対して個別のAWSアカウントを割り当てて権限分割やコストの明確な分離を行える
- VPCのオーナーはVPCレベルの要素(RouteTableやSubnetなど)を管理。利用者はSecurity Groupや自分が配置したリソースに責任を持つ
- 共有するアカウントは同一のAWS Organizationグループにある必要がある

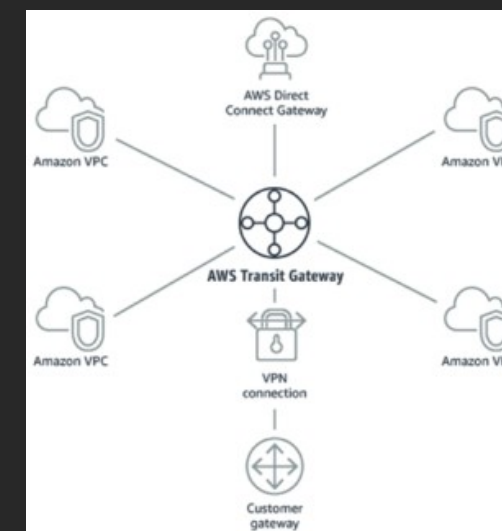
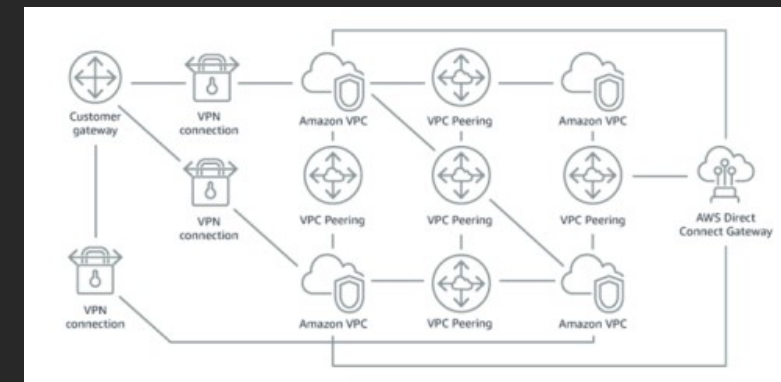


<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-sharing.html>

# 様々な拠点や複数のVPC間のネットワークを柔軟に構築する AWS Transit Gatewayを発表

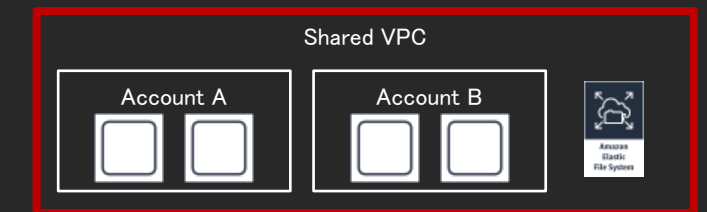
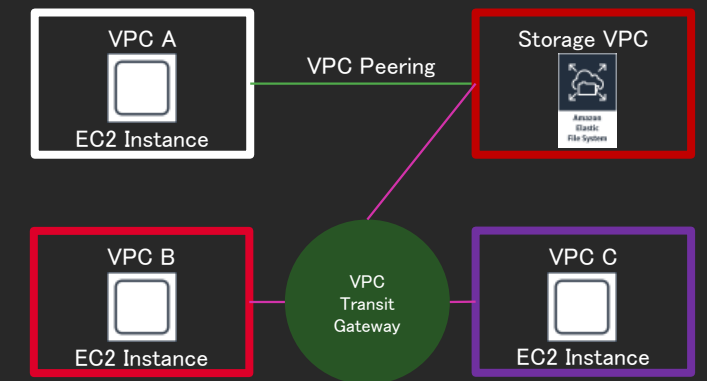


- 自分でソフトウェアルータを管理することなく、複数のVPCや拠点間を相互に接続・制御する
- CloudWatchによるメトリクス収集やVPC Flow Logs出力をサポート。SG/NACLによる通信制御も可能
- Transit Gatewayは冗長化がなされており、最大50Gbpsのバーストラフィックまで処理できる
- 現時点ではDirect ConnectをTGWにアタッチすることはできない。現在鋭意開発中
- 東京リージョンは現在準備中。課金体系は時間課金と処理データ量課金の2つの軸となる



# EFSが複数のアカウントやVPCからアクセス可能に

- EFSによるファイルシステムが、異なるアカウントや異なるVPCからアクセスできるようになり、1つのファイルシステムを共同利用しやすくなった
- VPC Peering/Transit Gatewayを経由してマウント可能。Shared VPCでも利用できる
- 細かな制約があるため、ドキュメントの確認を  
<https://docs.aws.amazon.com/efs/latest/ug/manage-fs-access-vpc-peering.html>
- 追加費用なしで利用可能。ピアリングやTGWの利用料・データ通信料は発生する



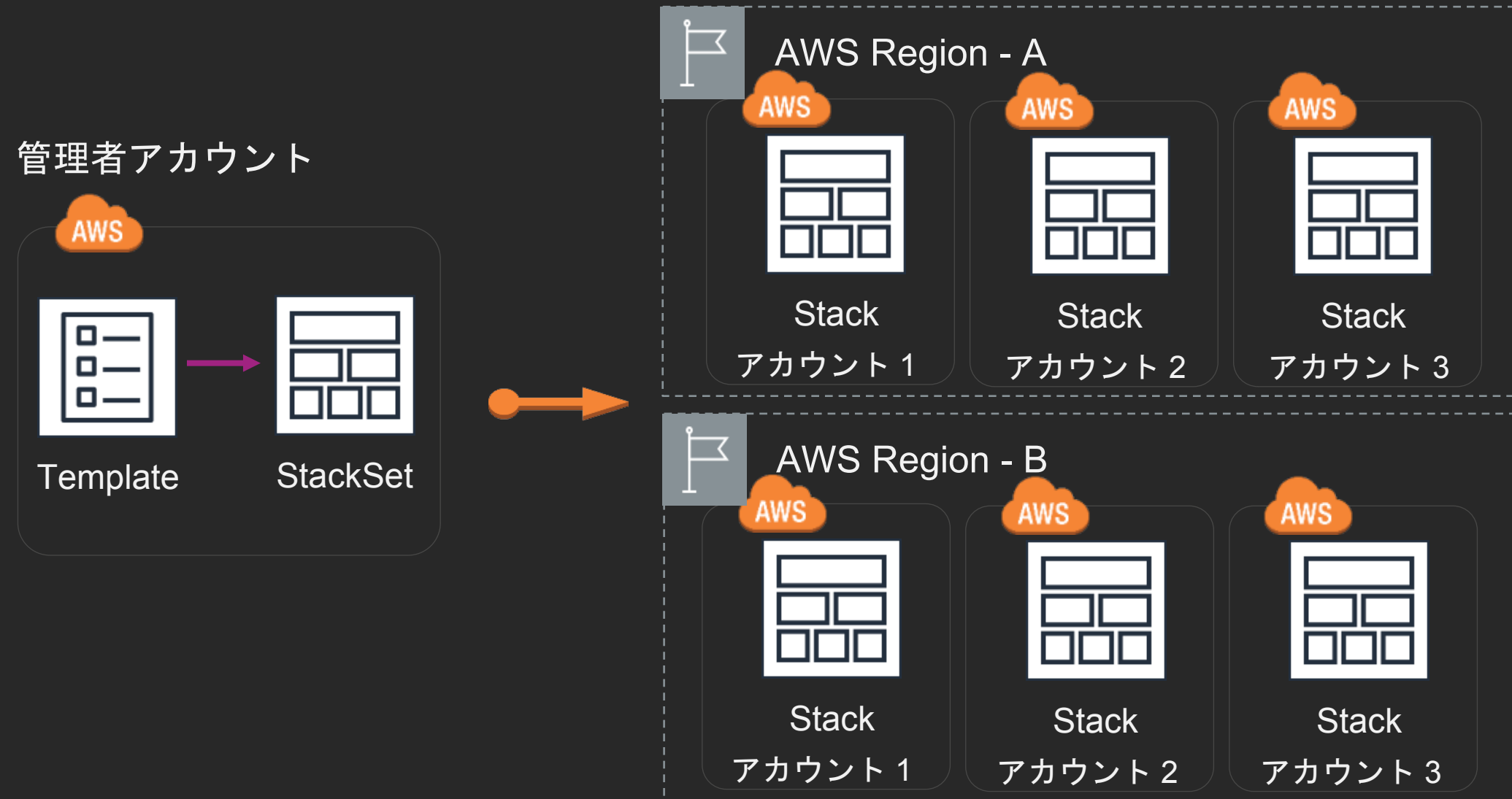
# マルチアカウント関連アップデート

アカウントレベルの払い出し管理がより容易に

1. アカウント作成 – Control Tower
2. リソース共有 – Resource Access Manager, Shared VPC
3. アクセス可能 – Transit GW, EFS
4. Provisioning – CFn StackSets
5. Monitoring – CloudWatch Events Bus
6. Audit – Security Hub, AWS Config, SSM Inventory
7. Remediation – SSM Automation

# CFn StackSets

- 一つのテンプレートを複数のAWSアカウント及び複数のリージョンに展開可能



# CloudWatch Event Bus

- 異なるアカウントに対して CloudWatch Event を発行
- アカウントごとに持つEvent Busを通じて配信
- 対象のEventについてCloudWatchルールを作成し、受け取るアカウントのEvent Busをアタッチ

イベントバス

デフォルトのイベントバスは、AWS のサービス、PutEvents API コール、デフォルトのイベントバスでアクセス許可を管理して、他のアカウントにアクセス許可のターゲットに追加することで、お客様とイベントを共有できます。

名前	説明
default	Default event bus

アクセス権限

アクセス許可の追加

タイプ	ID
企業	

CloudWatch  
ダッシュボード

ステップ 1: ルールの作成

AWS 環境で発生するイベントに基づいてターゲットを呼び出すためのルールを作成します。

イベントソース

イベントパターンを構築またはカスタマイズするか、スケジュールを設定してターゲットを呼び出します。

イベントパターン  スケジュール

サービス別のイベントに一致するイベントパターンの構築

サービス名: CodeBuild

イベントタイプ: すべてのイベント

このサービスからすべてのイベントに一致するイベントパターンを構築します。

イベントパターンのプレビュー

```
{
  "source": [
    "aws.codebuild"
  ]
}
```

ターゲット

イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに呼び出すターゲットを選択します。

別の AWS アカウントのイベントバス

アカウント ID\*: 123456789012

一致したイベントは、上記の AWS アカウントの「デフォルト」イベントバスで利用可能になります。

CloudWatch Events needs permission to send events to the 'default' event bus of the above AWS account. By continuing, you are allowing us to do so.

この特定のリソースに対して新しいロールを作成する

AWS\_Events\_Invoke\_Event\_Bus\_71458204

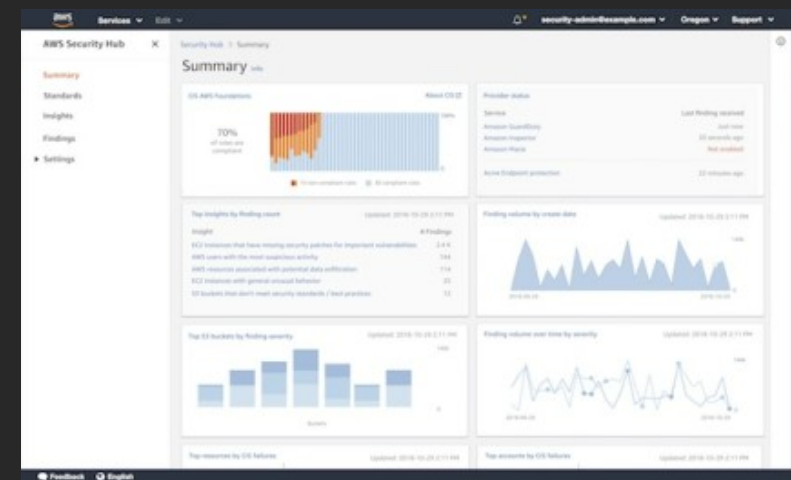
既存のロールの使用

CloudWatch Events のアイデンティティベースのポリシーの詳細については、[こちら](#) を参照してください。

ターゲットの追加\*

# 複数アカウントのセキュリティとコンプライアンスを管理 AWS Security Hubを発表

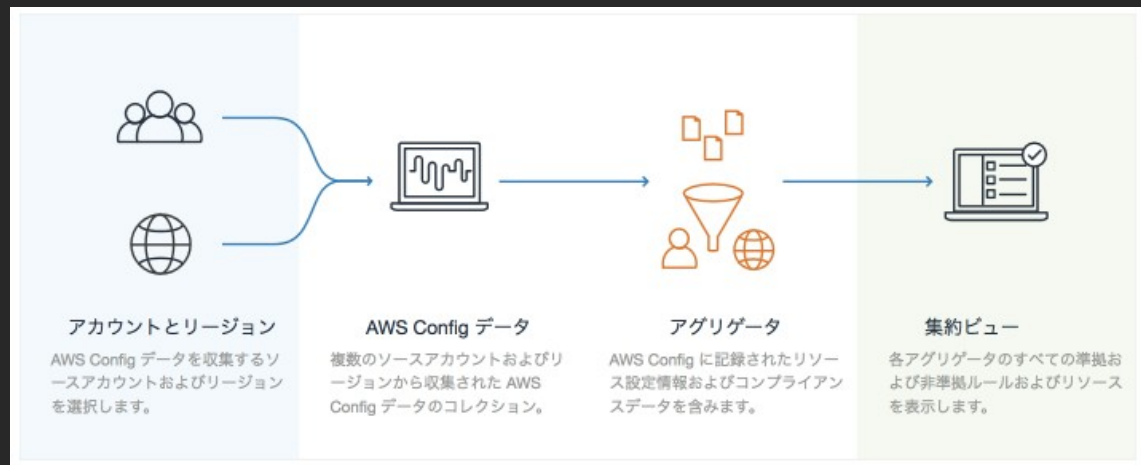
- アカウントの数が増えてくると、それぞれのアカウントでどういった事象が起きているかを管理することが難しくなってくる
- AWS Security Hubはそれぞれのアカウントでのセキュリティに関する留意事項を収集し、中央のアカウントで一括表示・対応ができるようにする
- アカウント毎に実施されるコンプライアンスチェックについても同様に対応できる
- 現在プレビュー期間につき追加コストは不要





# AWS Config Aggregator

- 複数アカウント、リージョンのConfig情報を集約



**AWS Config**

- ダッシュボード
- ルール
- リソース
- 設定
- 承認

---

- 集約ビュー
  - ルール
  - Resources**
  - アグリゲータ

---

最新情報

---

詳細情報

- ドキュメント
- パートナー
- よくある質問
- 料金表

Cost estimator

---

**Resources**

Search multi-account multi-region aggregator resources recorded by AWS Config. To view the details for each resource, click on the resource ID.

CloudFormation: Stack, EC2: Inst...

アグリゲータ: myOrg | リージョン: すべてのリージョン | アカウント: すべてのアカウント

Resource ID	Resource type	Region
arn:aws:cloudformation:ap-northeast-1:██████████:DXLab/5c-50d5ca9ff42a	CloudFormation Stack	ap-northeast-1
arn:aws:cloudformation:ap-northeast-1:██████████:EC2Insta-0-ddec-11e8-b7bc-50d5ca9ff4ae	CloudFormation Stack	ap-northeast-1
arn:aws:cloudformation:ap-northeast-1:██████████:Ethereum-Node/ea0a0110-a055-11e8-aa58-500c44f24ce6	CloudFormation Stack	ap-northeast-1
arn:aws:cloudformation:ap-northeast-1:██████████:██████████	CloudFormation Stack	ap-northeast-1

# AWS Systems Manager Inventory および Automation のマルチアカウント対応

- Inventory
  - OS上のソフトウェア構成を収集しAthenaでアカウントやリージョンをまたいで検索
- Automation
  - キュメントに定義した一連のAPI操作をアカウントやリージョンをまたいだAWSリソースに対して実行

The screenshot displays the AWS Systems Manager Inventory console. At the top, there are navigation tabs for 'ダッシュボード', '詳細ビュー', and '設定'. A blue information box states: 'この機能では、AWS Athena、AWS Glue、リソースデータの同期を使用してインベントリデータを表示します。この機能を使用するには、リソースデータの同期を選択する必要があります。料金が適用される場合があります。' Below this, there are buttons for 'リソースデータの同期', 'インベントリデータの表示', and 'リソースデータの同期の作成'. A search bar labeled '同期名によるフィルター' is present. A table shows synchronization details for 'ssm-resource' with a status of '成功' (Success). Below the table, there is a section for 'インベントリタイプ' set to 'AWS:Application', and buttons for 'CSVへエクスポート', 'Query History', and 'Run Advanced Queries'. A second search bar and a table with columns 'Package ID', 'Account ID', 'Region', and 'Publisher' are also visible.

同期名	同期日 (UTC)	前回のステータス	最終の同期 (UTC)	最後に成功した同期 (UTC)
ssm-resource	Wed, 11 Jul 2018 07:10:31 GMT	成功	Sun, 02 Dec 2018 22:46:05 GMT	Sun, 02 Dec 2018 22:46:05 GMT

# マルチアカウント関連アップデート

アカウントレベルの払い出し管理がより容易に

1. アカウント作成 – Control Tower
2. リソース共有 – Resource Access Manager, Shared VPC
3. アクセス可能 – Transit GW, EFS
4. Provisioning – CFn StackSets
5. Monitoring – CloudWatch Events Bus
6. Audit – Security Hub, AWS Config, SSM Inventory
7. Remediation – SSM Automation

# まとめ

# まとめ

1. DevOps と AWS Management Tools
2. AWS Management Toolsの概要
3. 各サービスの使い所と直近のアップデート
4. マルチアカウント管理

# re:Invent 2018 をキャッチアップするために

## 1. CFn と Management Tools セッションガイド

Amazon Web Services ブログ

re:Invent 2018 AWS CloudFormation セッションガイド

by AWS Japan Staff | on 26 NOV 2018 | in AWS CloudFormation, Management Tools | [Permalink](#) | [Share](#)



# AWS re:Invent 2018 ダイジェスト ～ AWS の最新動向を学ぶ～

AWS  
re:Invent ダイジェスト

～ AWS の最新動向を学ぶ～

2018年12月13日(木)  
サンライズビル大阪(大阪)にて開催

AWS  
re:Invent ダイジェスト

～ AWS の最新動向を学ぶ～

2018年12月11日(火)  
ベルサール汐留(東京)にて開催





# Thank you!