



【AWS Black Belt Online Seminar】

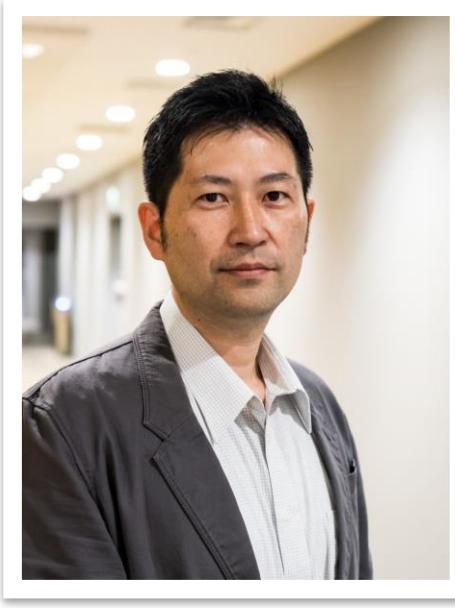
AWS CloudFormation

アマゾンウェブサービスジャパン株式会社

ソリューションアーキテクト 益子 直樹

2016.11.24

自己紹介



名前：益子 直樹（ましこ なおき）

所属：アマゾンウェブサービスジャパン
ソリューションアーキテクト

ロール：エンタープライズのお客様をご支援

経歴：通信キャリアで広域イーサやISPの
バックボーンの設計開発や運用を担当

好きなAWSサービス： CloudFormation
VPC
Direct Connect

内容についての注意点

- 本資料では2016年11月24日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。
- 価格は税抜表記となっています。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

Agenda

- AWS CloudFormationの概要
- スタック、テンプレート
- テンプレート作成ツール
- AWS CloudFormationにおける運用
- まとめ

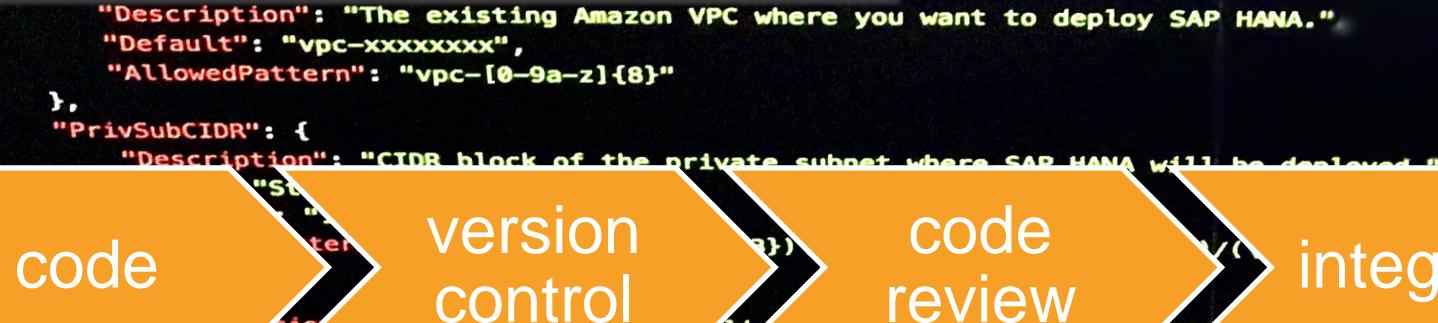


AWSを大規模で利用する時に、よくあるケース

- AWSリソースの管理・構築を効率化したい。
- 開発標準に基づいてインフラ作成・更新をしたい。
- リソースの依存関係やプロビジョニングの順序を確実にしたい。

そんなとき、「AWS CloudFormation」

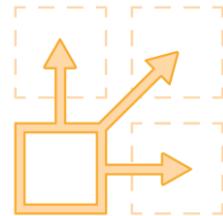
Infrastructure-as-code



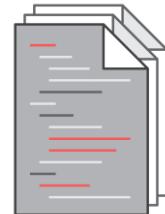
“インフラを全てソフトウェアで”

AWS CloudFormationとは

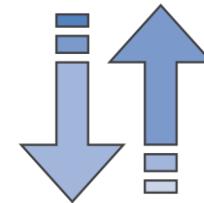
- EC2やELBといったAWSリソースの環境構築を、設定ファイル（テンプレート）を元に自動化できるサービス
- テンプレートを自由に作成できるため、自分好みのシステム構成を自動的に構築できる
- テンプレートには起動すべきリソースの情報をJSONやYAMLフォーマットのテキスト形式で記述する



テンプレートベースの
プロビジョニング



インフラを
コード化

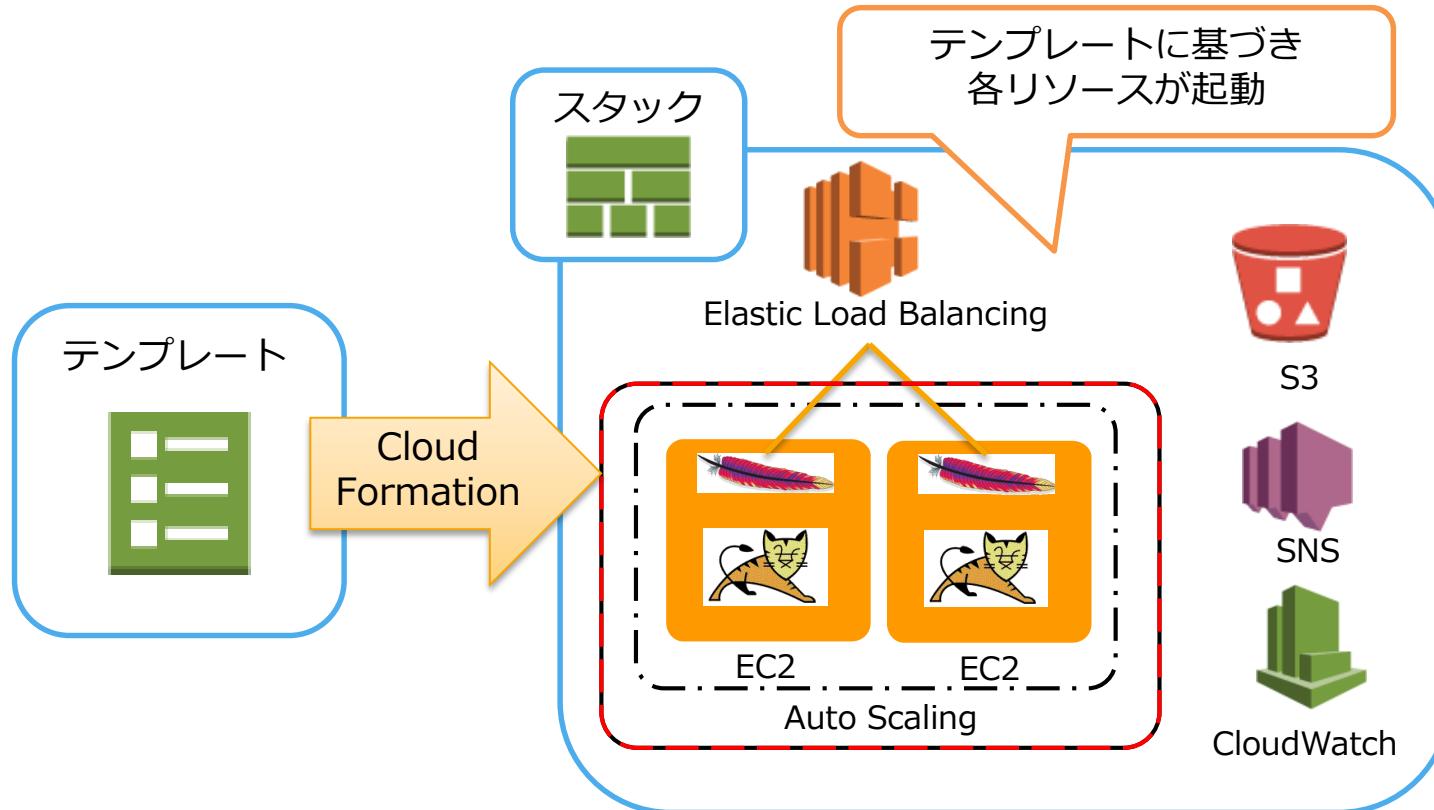


宣言・柔軟性



簡単に利用可能

AWS CloudFormationのイメージ



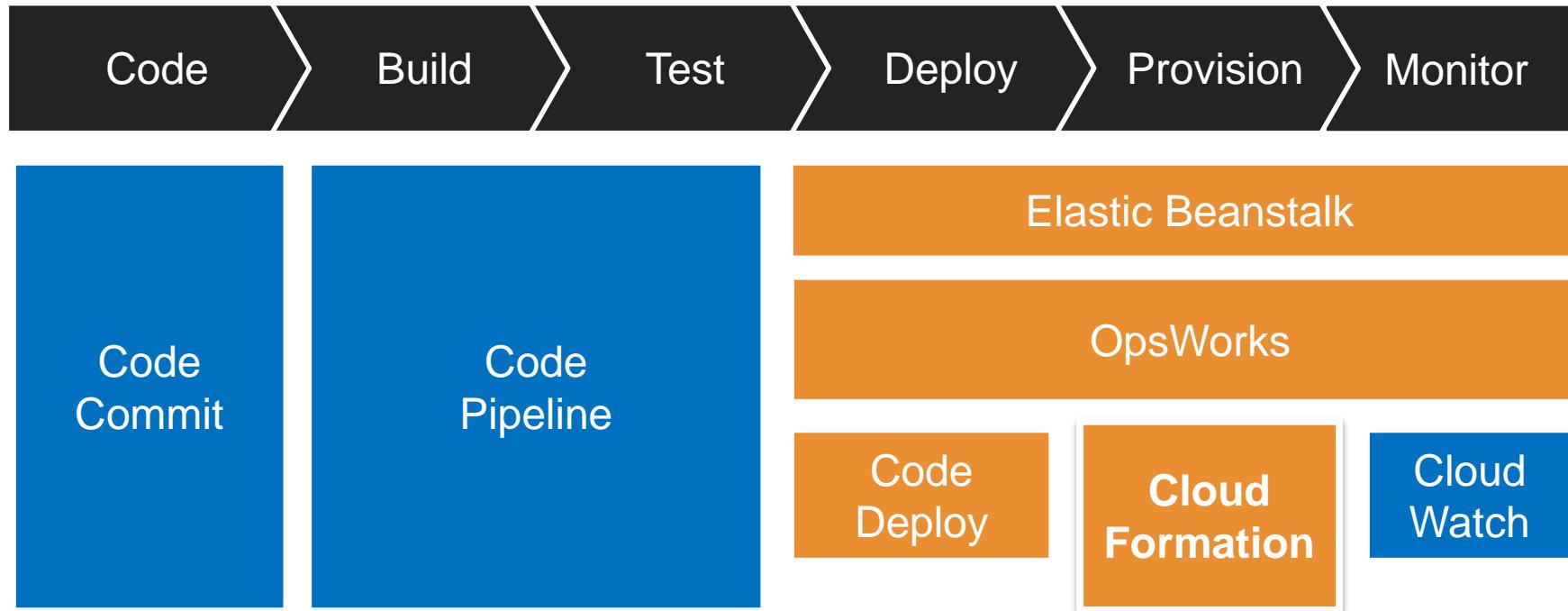
CloudFormationの特徴

- 一度テンプレートを作成すれば、同じ構成を再現できる
 - 開発環境の構築
 - Blogシステム、Webシステム、ゲームプラットフォームなど、同じ仕組みでアプリやデータが異なるようなもの
- ベストプラクティスが盛り込まれたテンプレートが使用可能
 - 複数のAZを跨いでリソースを配置する可用性の高い構成
 - セキュリティ要件を満たす上で必須のソフト・設定が入った構成
- 起動時にパラメータを渡せる
 - 例えばDBのエンドポイントをEC2に渡せる

利用料金

- AWS CloudFormationの利用自体は無料
- テンプレートに従って構築された各AWSサービスに対して課金
 - EC2インスタンス、ELB、RDSなど起動したリソースの費用が請求される

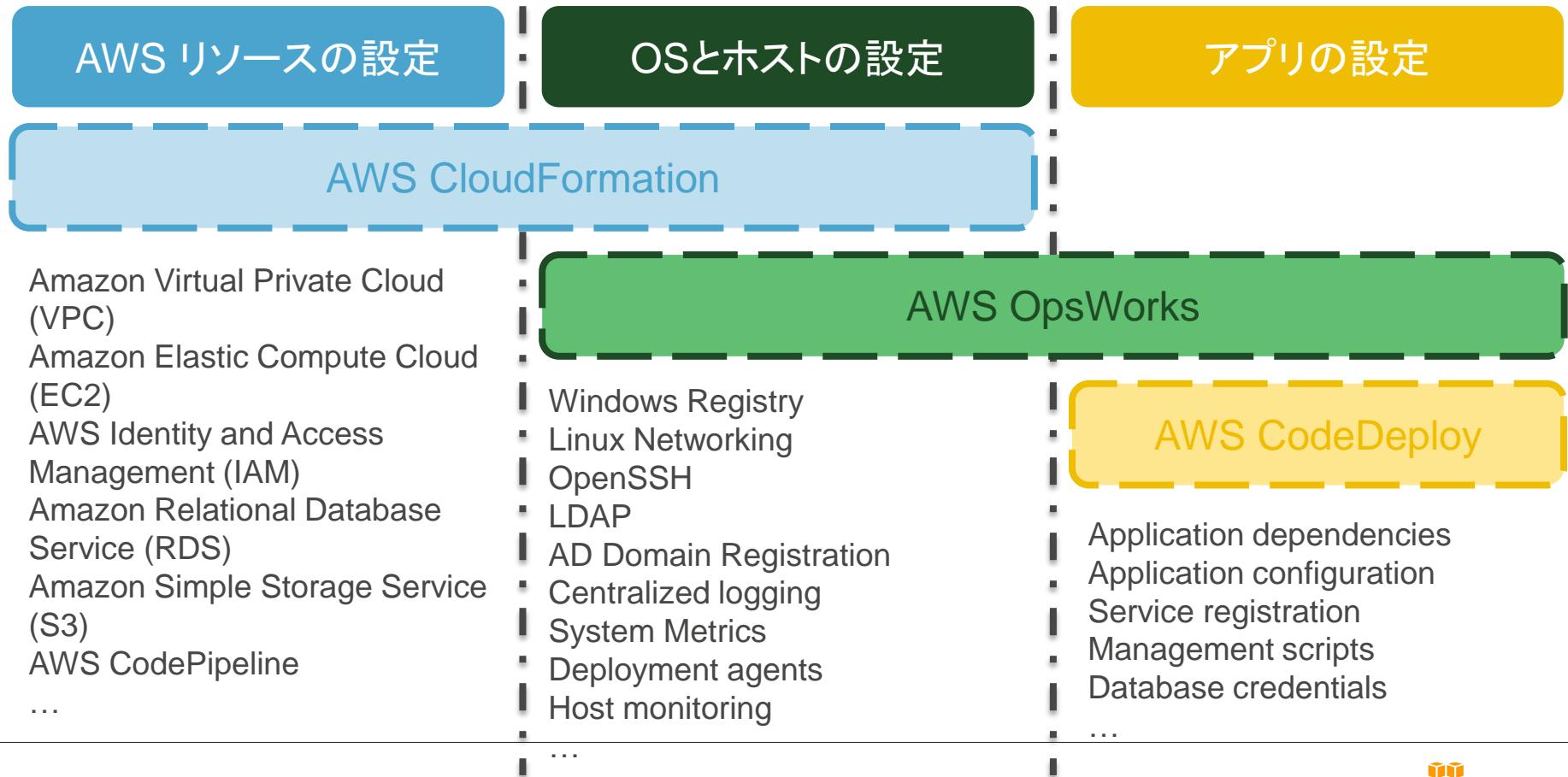
AWSから提供されているデプロイ＆マネージメント 関連サービス



AWSの構築・デプロイ自動化サービスの位置づけ



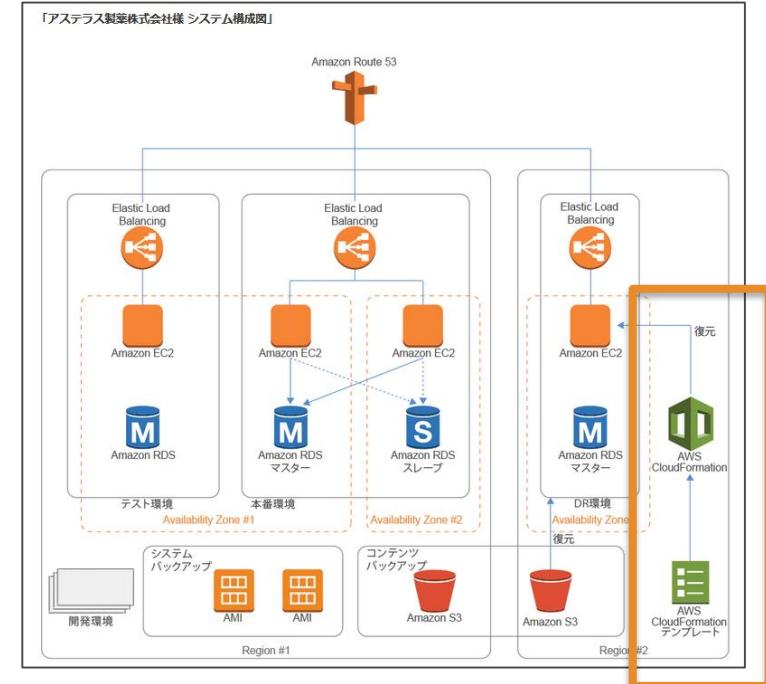
AWSサービスと設定するレイヤーの関係



AWS CloudFormationによるDR事例

アステラス製薬株式会社様

The screenshot shows the AWS Solutions Case Studies page for Astellas. The top navigation bar includes 'サインアップ', 'アカウント / コンソール', and '日本語'. Below the navigation is a search bar and links for 'サイト全体' (Search), '開発者' (Developer), and 'サポート' (Support). The main content area features the title 'AWS 導入事例: アステラス製薬株式会社' (AWS Case Study: Astellas Pharmaceutical Co., Ltd.) with the Astellas logo and tagline 'Leading Light for Life'. A large orange text box contains the message: '～AWS では、Webアクセス集中発生時の対応力も高く、信用性を維持することができます。～'. Below this, there is a section titled 'バックグラウンド' (Background) with detailed company information: '会社名: アステラス製薬株式会社', '所在地: 東京都中央区日本橋本町2-5-1', '創業年: 1923年', '資本金: 1,030億円 (2013年3月31日現在)', '従業員数: 17,454名 (2012年3月31日現在、連結ベース)', and '事業内容: 医薬品の製造・販売・輸出入'.



<http://aws.amazon.com/jp/solutions/case-studies/astellas/>

最近のアップデート

日付	主なアップデート
2016/11/18	AWS Serverless Application Model サポート
2016/10/6	クロススタック参照の拡張
2016/10/6	AWS CloudFormation サービスロール
2016/10/1	YAML サポート
2016/9/19	別のスタックのリソース出力を参照するスタックの作成が可能に
2016/8/11	ELBアプリケーションロードバランサーのリソースを追加
2016/7/20	AWS IoT 関連の6つのリソースを追加
2016/6/9	AWS::EC2::Flowlogの追加

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/ReleaseHistory.html>

AWS CloudFormationがサポートする主なサービス

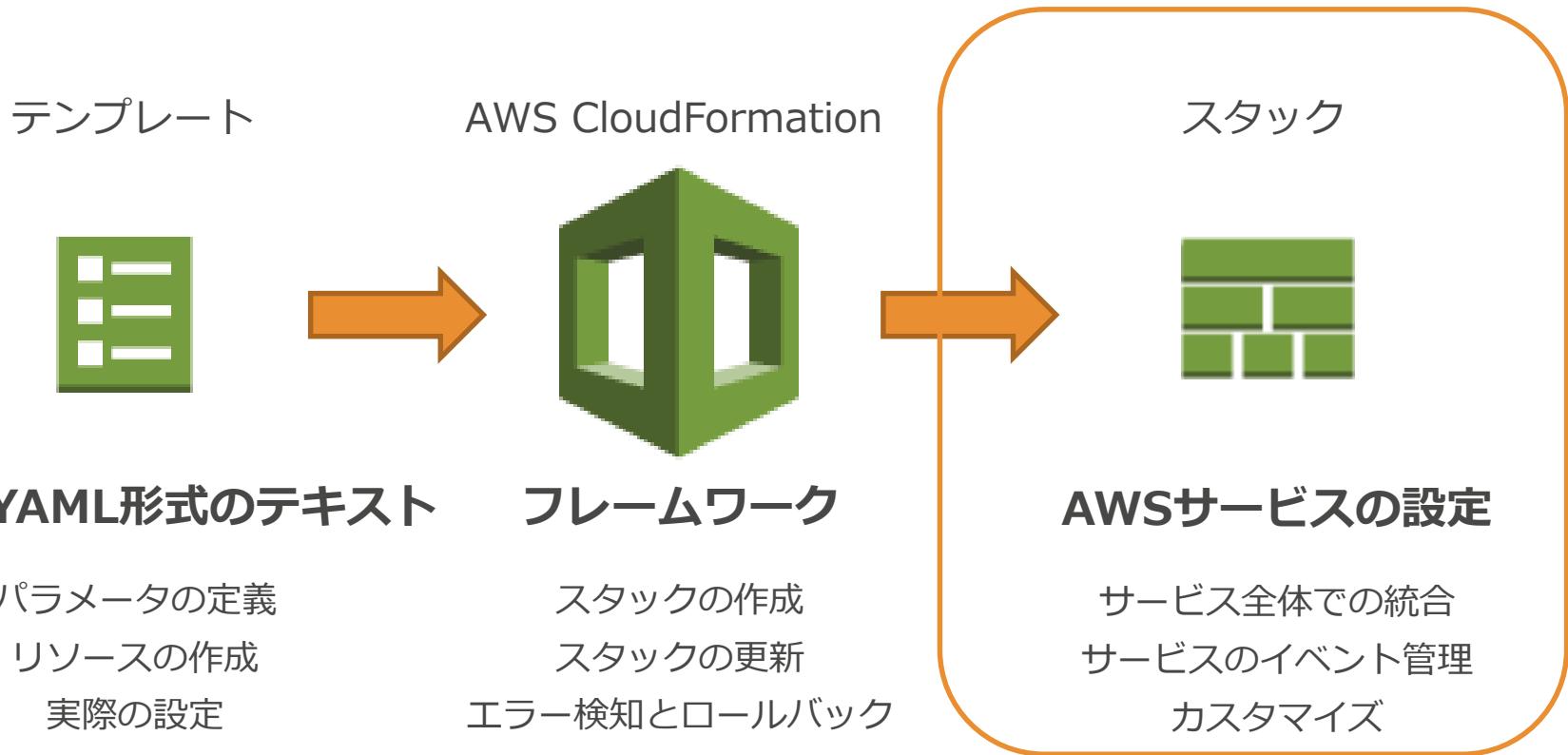
- Amazon EC2
- Amazon EC2 Container Service
- AWS Lambda (including event sources)
- Auto Scaling (including Spot Fleet)
- Amazon VPC
- Elastic Load Balancing (ALBも含む)
- Amazon Route 53
- Amazon CloudFront
- Amazon RDS
- Amazon Redshift
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon RDS for Aurora
- Amazon S3
- Amazon IoT
- AWS IAM (including managed policies)
- Simple AD
- Amazon Kinesis
- Amazon SNS
- Amazon SQS
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- AWS Data Pipeline
- AWS Elastic Beanstalk
- AWS OpsWorks
- AWS CodeDeploy
- Amazon WorkSpaces
- API Gateway

Agenda

- AWS CloudFormationの概要
- スタック、テンプレート
- テンプレート作成ツール
- AWS CloudFormationにおける運用
- まとめ

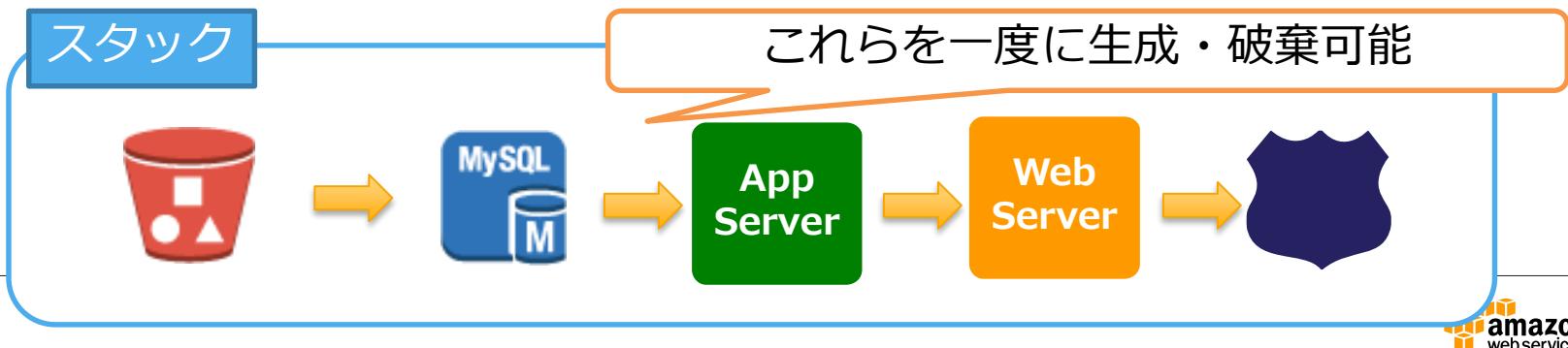


AWS CloudFormationとスタック、テンプレートの関係



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をスタックと呼ぶ
- 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



スタックの起動方法

- AWS Management Consoleから起動
- コマンドラインツールも利用可能
 - AWS Command Line Tool : <http://aws.amazon.com/cli/>
- 各種SDKでもサポート
 - Java : <http://aws.amazon.com/jp/sdkforjava/>
 - .NET : <http://aws.amazon.com/jp/sdkfornet/>
 - PHP : <http://aws.amazon.com/jp/sdkforphp/>
 - Ruby : <http://docs.aws.amazon.com/AWSSdkDocsRuby/latest/DeveloperGuide/ruby-dg-setup.html>
 - JavaScript : <http://aws.amazon.com/documentation/sdkforjavascript/>
 - Python : <http://docs.pythonboto.org/en/latest/>
 - GO: <https://docs.aws.amazon.com/sdk-for-go/api/service/cloudformation/>

AWS Management Consoleによる起動-1

The screenshot shows the AWS Management Console with the CloudFormation service selected. The top navigation bar includes links for AWS, Services, and various AWS products like Lambda, S3, and CloudWatch. The user is Yoichi Takizawa. The main content area is titled 'Create Stack' and displays three sections:

- Design a template**: Describes CloudFormation templates and how to use the AWS CloudFormation Designer. A 'Design template' button is present.
- Create a Stack**: Describes deploying infrastructure resources and applications. It states there are no stacks and provides a 'Create New Stack' button, which is highlighted with a red rectangle.
- Create a Template from your Existing Resources**: Describes using CloudFormer to capture existing resources. It includes a 'Launch CloudFormer' button.

AWS Management Consoleによる起動-2

The screenshot shows the 'Select Template' step in the AWS CloudFormation console. The user has entered 'MyStack' in the 'Name' field. The 'Source' section offers three options: 'Select a sample template' (radio button selected), 'Upload a template to Amazon S3' (with a 'Browse...' button and a message 'No file selected.'), and 'Specify an Amazon S3 template URL' (with a text input field containing 'https://s3.amazonaws.com/cloudformation-samples-us-east-1/'). A large orange callout box points to the 'Name' field with the text 'Stackの名前を入力'. Another orange callout box points to the 'Template' section with the text 'テンプレートを指定する。3種類の方法から選択'. Three smaller orange callout boxes point to each of the 'Source' options with their respective descriptions: 'サンプルを使う場合はこちら', '手持ちのテンプレートを使う場合はこちらを選択しファイルを指定', and 'S3に配置済みのテンプレートを指定する場合はこちら'.

Services ▾ Edit ▾

Masato Kobayashi ▾ Tokyo ▾ Help ▾

Select Template

Specify Parameters

Options

Review

Stack

Specify a stack name and then select the template that describes the stack that you want to create.

An AWS CloudFormation stack is a collection of related resources that you provision and manage together.

Name MyStack

Template

A template is a JSON-formatted file that describes the resources in your stack. You can either select a sample template or upload a template from an Amazon S3 bucket.

Source

Select a sample template

Inosho Social Networking Platform

Upload a template to Amazon S3

No file selected.

Specify an Amazon S3 template URL

https://s3.amazonaws.com/cloudformation-samples-us-east-1/

Cancel Next

Stackの名前を入力

テンプレートを指定する。
3種類の方法から選択

サンプルを使う場合はこちら

手持ちのテンプレートを使う場合はこちらを選択しファイルを指定

S3に配置済みのテンプレートを指定する場合はこちら

AWS Management Consoleによる起動-3

The screenshot shows the 'Specify Parameters' step of an AWS launch template wizard. The left sidebar has 'Specify Parameters' selected. The main area shows two parameters: 'DBRootPassword' set to '*****' and 'InstanceType' set to 'm1.small'. A callout bubble points to the 'DBRootPassword' field with the text 'テンプレートによってはパラメータの入力が必要(後述)' (Input required for parameters depending on the template (see later)). At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

Specify Parameters

Specify values or use the default values for the parameter.

Parameters

DBRootPassword Root password for MySQL

InstanceType WebServer EC2 instance type

Cancel Previous Next

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Feedback

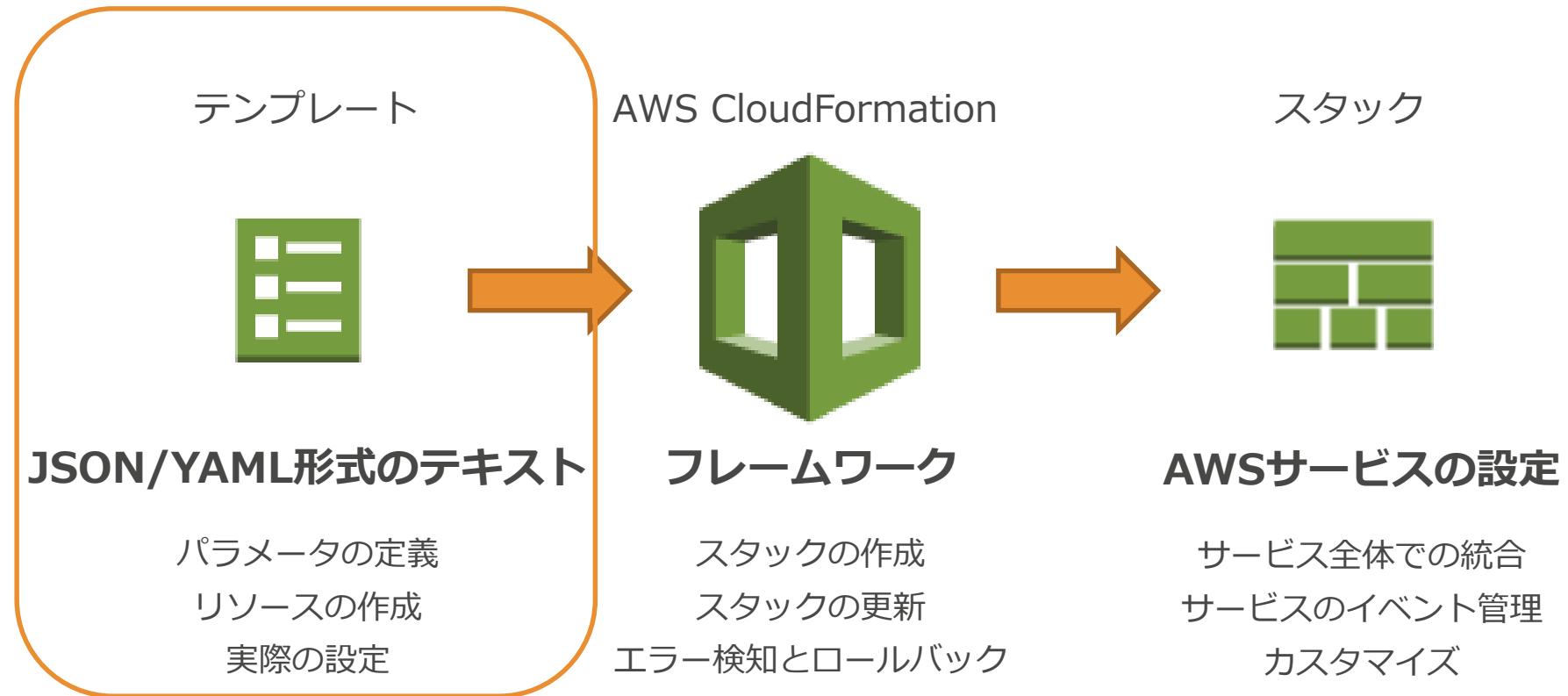
AWS Management Consoleによる起動-4

Stack Name: MyStack | Created Time: 2014-07-06 15:57:52 UTC+0900 | Status: CREATE_COMPLETE | Description: AWS CloudFormer Beta - template creation prototype application.

Outputs

Key	Value	Description
URL	http://ec2-54-92-45-141.ap-northeast-1.compute.amazonaws.com	AWS CloudFormer Prototype URL. Use this endpoint ...

AWS CloudFormationとテンプレート、スタックの関係



CloudFormationの利用パターン

クイックスタート

- セキュリティや可用性を考慮したベストプラクティス
- マイクロソフト、SAP、トレンドマイクロなどの製品をサポート
- 15分程度でアプリケーションをAWS上に展開可能。カスタマイズも可能

サンプルコード & テンプレート

- これからCloudFormationを使い始める場合
- 独自のテンプレートを短時間で作成したい場合

自らテンプレートを作成

- JSON/YAML形式でテンプレートを記述
- CloudFormerを利用する
- CloudFormation Designerを利用する

導入の容易さ

フレキシブル

クイックスタートリファレンス

メニュー  製品 ▾ ソリューション 料金表 お客様事例 よくあるご質問 トレーニング ソフトウェア 詳細 ▾ 日本語 ▾ アカウント ▾ サインアップ

AWS クイックスタート

AWS で代表的なデプロイメントを自動化

開発運用

- コンプライアンス
- Microsoft ワークロード
- Oracle ワークロード
- SAP ワークロード
- ネットワーキングとリモートアクセス
- ストレージ
- セキュリティ
- マーケティングとコマース
- ビッグデータ

このページのクイックスタートは、セキュリティ AWS ソリューション設計者によって構築されました。このページのクイックスタートは、セキュリティ AWS ソリューション設計者によって構築されました。このページのクイックスタートは、セキュリティ AWS ソリューション設計者によって構築されました。

コミュニティのクイックスタート | ? よくあるご質問

クティスに沿って、
イでは、重要なテクノロジ
環境をいくつかの簡単な
手順で構築して、すぐに使用を始めることができます。

ご質問がありましたら、お知らせください。

開発運用

 Puppet

AWS の宣言型のモデルベース構成管理ソリューション

[詳細](#)

[ガイドを表示](#) | [デプロイ](#)

 Chef Server

インフラストラクチャ管理のための Chef DevOps プラットフォームの基盤

[詳細](#)

[ガイドを表示](#) | [デプロイ](#)

 Ansible Tower

クラウドのプロビジョニング、構成管理、IT タスクのためのソリューション

[詳細](#)

[ガイドを表示](#)

 PowerShell DSC

Microsoft Windows 用のタスクの自動化とシステム構成

[詳細](#)

[ガイドを表示](#)

 Microsoft

AWS の継続的インテグレーション/継続的配信 (CI/CD) パイプラインのための AWS アーキテクチャ

[詳細](#)

[ガイドを表示](#)

<https://aws.amazon.com/jp/quickstart/>

クイックスタートリファレンスー SAP HANA

AWS Quick Starts

SAP HANA on AWS

With X1 instances certified by SAP for your production workloads

Deploy on AWS into a new VPC
(or deploy into an existing VPC)
Requires SAP media and RHEL subscription - see guide

View guide – HTML | PDF

Amazon S3
HANA media bucket
Backup snapshots

VPC CIDR (e.g., 10.0.0.0/16)

Private subnet (e.g., 10.0.1.0/24)

Public subnet (e.g., 10.0.2.0/24)

RDP instance
HANA Studio

VPC NAT gateway

Bastion instance
Inbound SSH
Elastic IP

router

Internet gateway

Internet

This Quick Start helps you deploy fully functional SAP HANA systems on the AWS Cloud, following best practices from AWS and SAP. The Quick Start ensures that AWS services and the OS (SLES or RHEL) are optimally configured to achieve the best performance for your SAP HANA system.

For additional Quick Starts, [see the complete catalog](#).

“Using AWS SAP HANA Quick Start, we only needed to push a few buttons to get a functioning SAP HANA solution.”
Philip Miller, Director
Brooks Brothers

You can request up to \$2,000 in AWS credit for an SAP HANA POC.

Amazon webservices

AWS re:Invent Products Solutions Pricing Software Support Customers More English My Account Sign In to the Console

ここをクリックすると CloudFormationの画面が起動される

クイックスタートリファレンスー SAP HANA

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The top navigation bar includes 'サービス', 'リソースグループ', and various icons. The current step is 'CloudFormation > Stacks > Create Stack'. The left sidebar has tabs: 'Select Template' (highlighted), 'Specify Details', 'Options', and 'Review'. The main content area is titled 'Select Template' with the sub-instruction: 'Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.' It offers three options: 'Design a template' (using AWS CloudFormation Designer), 'Choose a template' (uploading from S3 or specifying a URL), and 'Specify an Amazon S3 template URL' (which is selected). A note indicates that the URL must be a valid JSON/YAML file. A large orange callout bubble points to the 'Specify an Amazon S3 template URL' section with the text: 'S3にアップされているテンプレートが指定される'. At the bottom right are 'Cancel' and 'Next' buttons.

S3にアップされているテンプレートが指定される

Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

Design a template Use AWS CloudFormation Designer to create or modify an existing template. [Learn more.](#)

Design template

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)

Select a sample template

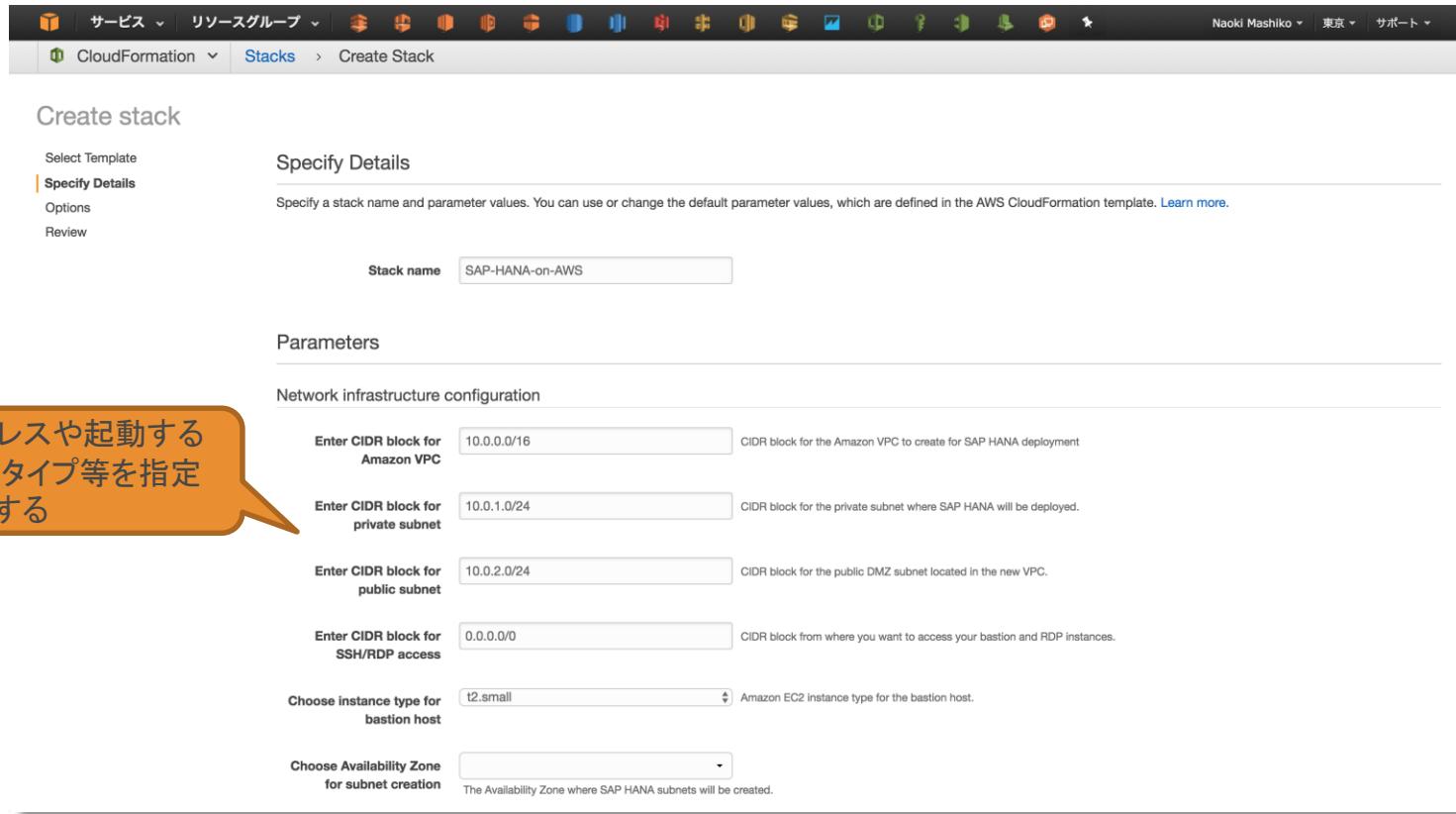
Upload a template to Amazon S3
ファイルを選択 選択されていません

Specify an Amazon S3 template URL

<https://s3.amazonaws.com/quickstart-reference/sap/hana/latest/template> [View/Edit template in Designer](#)

Cancel Next

クイックスタートリファレンスー SAP HANA



CIDRのアドレスや起動するインスタンスタイプ等を指定する

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name: SAP-HANA-on-AWS

Parameters

Network infrastructure configuration

Enter CIDR block for Amazon VPC: 10.0.0.0/16

Enter CIDR block for private subnet: 10.0.1.0/24

Enter CIDR block for public subnet: 10.0.2.0/24

Enter CIDR block for SSH/RDP access: 0.0.0.0/0

Choose instance type for bastion host: t2.small

Choose Availability Zone for subnet creation: [Select]

The Availability Zone where SAP HANA subnets will be created.

CloudFormationの利用パターン

クイックスタート

- セキュリティや可用性を考慮したベストプラクティス
- マイクロソフト、SAP、トレンドマイクロなどの製品をサポート
- 15分程度でアプリケーションをAWS上に展開可能。カスタマイズも可能

サンプルコード & テンプレート

- これからCloudFormationを使い始める場合
- 独自のテンプレートを短時間で作成したい場合

自らテンプレートを作成

- JSON/YAML形式でテンプレートを記述
- CloudFormerを利用する
- CloudFormation Designerを利用する

導入の容易さ

フレキシブル

AWS CloudFormationのサンプル



AWS CloudFormation は、AWS でのプロビジョニングと管理を単純化します。目的のサービスまたはアプリケーションのアーキテクチャに合わせたテンプレートを作成して、AWS CloudFormation でそのテンプレートを使用すると、そのサービスまたはアプリケーション（「スタック」といいます）のプロビジョニングをすばやく確実に行うことができます。必要に応じてスタックを更新または複製することも簡単です。

ここで紹介するサンプルテンプレートは、これから AWS CloudFormation を使い始める場合や、独自のテンプレートを短時間で作成したいという場合に役立ちます。

AWS サービスによるテンプレートとスニペット

AWS サービスによるサンプルテンプレートの一覧を表示します。

AWS サービスによるテンプレートスニペットの一覧を表示します。

開発者ドキュメントに、他の例とリファレンスがあります。

アプリケーションフレームワーク

アプリケーションフレームワークテンプレートは、LAMP や Ruby on Rails などのよく使われるフレームワークを、AWS CloudFormation を使用してプロビジョニングする方法を示すものです。

実験的/コミュニティのサンプル

AWS Labs では、実験的な AWS CloudFormation テンプレートなど、AWS の従業員やパートナーが参加しているプロジェクトを紹介しています。

リファレンス実装

AWS Quick Start は、Microsoft Windows Server や SAP HANA などの一般的な IT ワークロードについて、AWS CloudFormation テンプレートと詳細なデプロイメントガイドをまとめたものです。

AWS Test Drive では、AWS CloudFormation を使用してプロビジョニングされた、Oracle や Microsoft などのベンダーからの IT ソリューションをプライベートのサンドボックス環境で試用することができます。

サンプルソリューション

サンプルのソリューションテンプレートは、一般的なアプリケーションが含まれるエンドツーエンドのソリューションを作成する方法を示すものです。AWS は、これらのサンプル内のアプリケーションのサポートや保守は行いません。これらのサンプルは、AWS CloudFormation の能力を示すことのみを目的としています。

注: AWS CloudFormation サンプルテンプレートは AWS カスタマーアグリーメントが適用される AWS コンテンツであり、AWS CloudFormation との関連でのみ使用が許可されます。

AWS MARKETPLACE

AWS で人気のソフトウェアをワンクリックで起動



AMAZON EC2

12/1 から Amazon EC2 が最大 25% 値下げ。C4、M4、T2 が東京リー



<https://aws.amazon.com/jp/cloudformation/aws-cloudformation-templates/>



テンプレートとスニペット

- テンプレートのサンプルを利用可能

<http://aws.amazon.com/jp/cloudformation/aws-cloudformation-templates/>

Amazon EC2

テンプレート名	説明	表示	デザイナーで表示	作成する
セキュリティグループの Amazon EC2 インスタンス	Amazon EC2 セキュリティグループ内に Amazon EC2 インスタンスを作成します。	表示	デザイナーで表示	Launch Stack
Elastic IP アドレスを持つ Amazon EC2 インスタンス	Amazon EC2 インスタンスを作成し、Elastic IP アドレスをインスタンスに関連付けます。	表示	デザイナーで表示	Launch Stack
エフェメラルドライブが指定された Amazon EC2 インスタンス	ブロックデバイスマッピングを使用して、エフェメラルドライブが指定された Amazon EC2 インスタンスを作成します。	表示	デザイナーで表示	Launch Stack

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  
  "Description": "AWS CloudFormation Sample Template  
EC2InstanceWithSecurityGroupSample: Create an Amazon EC2 instance running the Amazon  
Linux AMI. The AMI is chosen based on the region in which the stack is run. This example  
creates an EC2 security group for the instance to give you SSH access. **WARNING** This  
template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you  
create a stack from this template.",  
  
  "Parameters": {  
    "KeyName": {  
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instance",  
      "Type": "AWS::EC2::KeyPair::KeyName",  
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."  
    },  
  
    "InstanceType": {  
      "Description": "WebServer EC2 instance type",  
      "Type": "String",  
      "Default": "t2.small",  
      "AllowedValues": [ "t1.micro", "t2.nano", "t2.micro", "t2.small", "t2.medium", "t2.large",  
        "m1.small", "m1.medium", "m1.large", "m1.xlarge", "m2.xlarge", "m2.2xlarge", "m2.4xlarge",  
        "m3.medium", "m3.large", "m3.xlarge", "m3.2xlarge", "m4.large", "m4.xlarge", "m4.2xlarge",  
        "m4.4xlarge", "m4.10xlarge", "c1.medium", "c1.large", "c3.large", "c3.xlarge", "c3.2xlarge",  
        "c3.4xlarge", "c3.8xlarge", "c4.large", "c4.xlarge", "c4.2xlarge", "c4.4xlarge", "c4.8xlarge",  
        "g2.2xlarge", "g2.8xlarge", "r3.large", "r3.xlarge", "r3.2xlarge", "r3.4xlarge", "r3.8xlarge",  
        "i2.xlarge", "i2.2xlarge", "i2.4xlarge", "i2.8xlarge", "d2.xlarge", "d2.2xlarge", "d2.4xlarge",  
        "d2.8xlarge", "hi1.4xlarge", "hs1.8xlarge", "cr1.8xlarge", "cc2.8xlarge", "cg1.4xlarge" ]  
    }  
  },  
  
  "Resources": {  
    "AutoScalingGroup": {  
      "Type": "AWS::AutoScaling::AutoScalingGroup",  
      "Properties": {  
        "LaunchConfigurationName": "LaunchConfigForEC2Instance",  
        "MinSize": 1,  
        "MaxSize": 1,  
        "DesiredCapacity": 1,  
        "VPCZoneIdentifier": "vpc-00000000",  
        "HealthCheckGracePeriod": 300,  
        "HealthCheckType": "ELB",  
        "Cooldown": 300,  
        "TerminationPolicies": "Default",  
        "MetricsCollection": {  
          "Metrics": ["CPUUtilization"],  
          "Granularity": "OneMinute",  
          "SampleWindow": 60  
        },  
        "Tags": [{  
          "Key": "Name",  
          "Value": "MyAutoScalingGroup"  
        }]  
      }  
    },  
  
    "LaunchConfig": {  
      "Type": "AWS::CloudFormation::Designer",  
      "Properties": {  
        "Template": {  
          "Resources": {  
            "1": {  
              "Type": "AWS::EC2::Instance",  
              "Properties": {  
                "ImageId": "ami-00000000",  
                "InstanceType": "t2.small",  
                "SecurityGroups": ["InstanceSecurityGroup"],  
                "KeyName": "MyKey",  
                "BlockDeviceMappings": [{  
                  "DeviceName": "/dev/sda1",  
                  "VirtualName": "/dev/xvda",  
                  "Ebs": {  
                    "VolumeSize": 8, "DeleteOnTermination": true  
                  },  
                  "NoDevice": false  
                }]  
              }  
            }  
          }  
        }  
      }  
    },  
  
    "SecurityGroup": {  
      "Type": "AWS::EC2::SecurityGroup",  
      "Properties": {  
        "GroupDescription": "Allow SSH access",  
        "GroupType": "VPC",  
        "VpcId": "vpc-00000000",  
        "IpPermissions": [{  
          "IpProtocol": "tcp",  
          "FromPort": 22, "ToPort": 22, "IpRanges": [{"CidrIp": "0.0.0.0/0"}]  
        }]  
      }  
    },  
  
    "Instance": {  
      "Type": "AWS::EC2::Instance",  
      "Properties": {  
        "ImageId": "ami-00000000",  
        "InstanceType": "t2.small",  
        "SecurityGroups": ["InstanceSecurityGroup"],  
        "KeyName": "MyKey",  
        "BlockDeviceMappings": [{  
          "DeviceName": "/dev/sda1",  
          "VirtualName": "/dev/xvda",  
          "Ebs": {  
            "VolumeSize": 8, "DeleteOnTermination": true  
          },  
          "NoDevice": false  
        }]  
      }  
    }  
  },  
  
  "Outputs": {  
    "AutoScalingGroupURL": {  
      "Description": "The URL of the Auto Scaling group",  
      "Value": {  
        "Ref": "AutoScalingGroup"  
      }  
    },  
  
    "LaunchConfigURL": {  
      "Description": "The URL of the Launch Configuration",  
      "Value": {  
        "Ref": "LaunchConfig"  
      }  
    }  
  }  
}
```

The screenshot shows the AWS CloudFormation Designer interface. At the top, there's a navigation bar with tabs like 'Services', 'Resource Groups', 'Stacks', 'Components', and 'Template'. Below the navigation is a search bar and a 'Close' button. The main area has a title 'File: Template1'. On the left, there's a sidebar titled 'Resource types' with a tree view of available services: ApplicationAutoScaling, APIGateway, AutoScaling, CertificateManager, CloudFormation, CloudFront, CloudTrail, CloudWatch, CodeCommit, CodeDeploy, CodePipeline, and Config. The 'CloudFormation' node is expanded. In the center, there's a diagram showing an 'EC2Instance' (represented by an orange square) connected to a 'SecurityGroup' (represented by a red square with a lock icon). A 'LaunchConfig' (represented by a blue square) is also shown, with arrows indicating its connection to both the instance and the security group. At the bottom, there are tabs for 'Properties', 'Metadata', 'CreationPolicy', 'DeletionPolicy', 'DependsOn', and 'Condition'. The 'Properties' tab is currently selected, showing the configuration for the EC2 instance.

テンプレートとスニペット

テンプレート

- [Auto Scaling](#)
- [Amazon DynamoDB](#)
- [Amazon EC2](#)
- [Amazon ElastiCache](#)
- [AWS Elastic Beanstalk](#)
- [Elastic Load Balancing](#)
- [AWS Identity and Access Management](#)
- [AWS OpsWorks](#)
- [Amazon Relational Database Service](#)
- [Amazon Redshift](#)
- [Amazon Route 53](#)
- [Amazon Simple Storage Service](#)
- [Amazon Simple Queue Service](#)
- [Amazon Virtual Private Cloud](#)

テンプレートスニペット

- 一般的なテンプレートスニペット
- [Auto Scaling テンプレートスニペット](#)
- [AWS CloudFormation テンプレートスニペット](#)
- [Amazon CloudFront のテンプレートスニペット](#)
- [Amazon CloudWatch テンプレートスニペット](#)
- [Amazon CloudWatch Logs テンプレートスニペット](#)
- [Amazon EC2 テンプレートスニペット](#)
- [Amazon EC2 Container Service テンプレートスニペット](#)
- [Amazon Elastic File System サンプルテンプレート](#)
- [Elastic Beanstalk テンプレートスニペット](#)
- [Elastic Load Balancing テンプレートのスニペット](#)
- [AWS Identity and Access Management のテンプレートスニペット](#)
- [AWS Lambda テンプレート](#)
- [AWS OpsWorks テンプレートスニペット](#)
- [Amazon Redshift テンプレートスニペット](#)
- [Amazon RDS テンプレートスニペット](#)
- [Amazon Route 53 テンプレートスニペット](#)
- [Amazon S3 テンプレートスニペット](#)
- [Amazon SNS テンプレートスニippet](#)
- [Amazon SQS テンプレートスニippet](#)

Serverless Application Model (SAM)

- より簡単にAWS上でサーバレスアプリケーションを管理および展開するもの
- CloudFormationでネイティブにサポートされている構文を使用して、サーバレスアプリケーションとして代表的なサービスであるLambda、API Gateway、DynamoDBを簡単に定義可能
- 利用にはCloudFormationにてサーバレスアプリケーションの宣言を行う

YAML

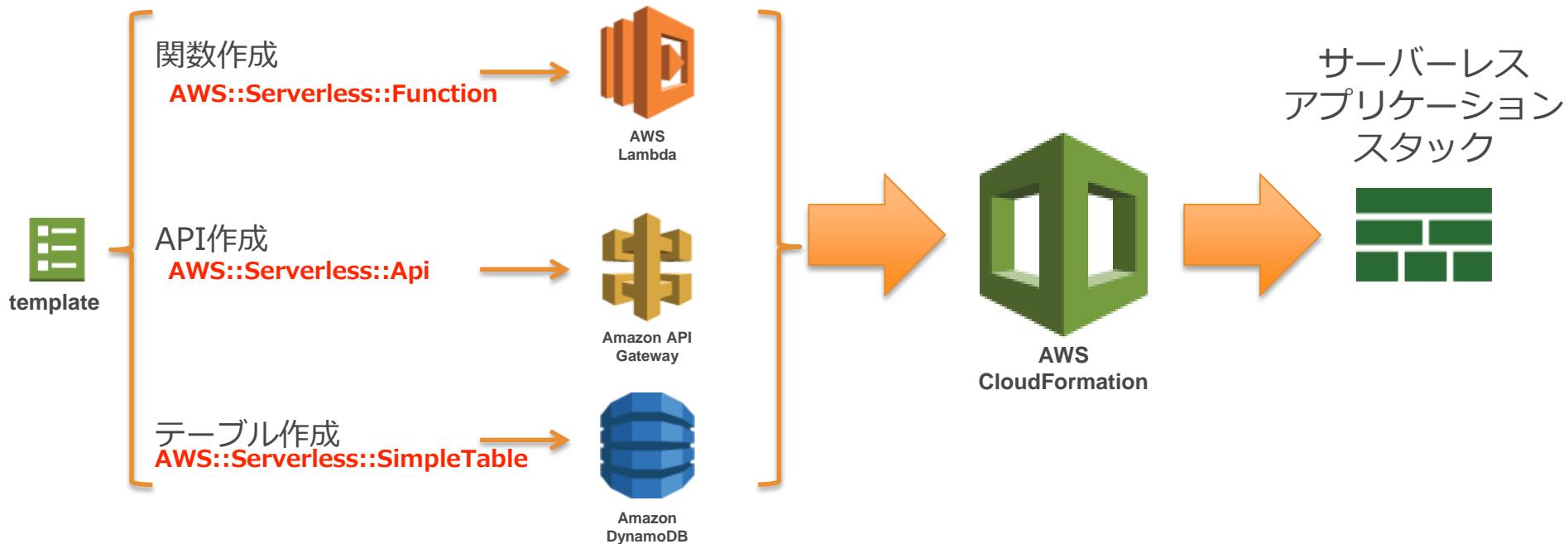
```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31
```

CloudFormationのテンプレート内に、サーバレスアプリケーションを定義していることを通知する

New

Serverless Application Model (SAM)

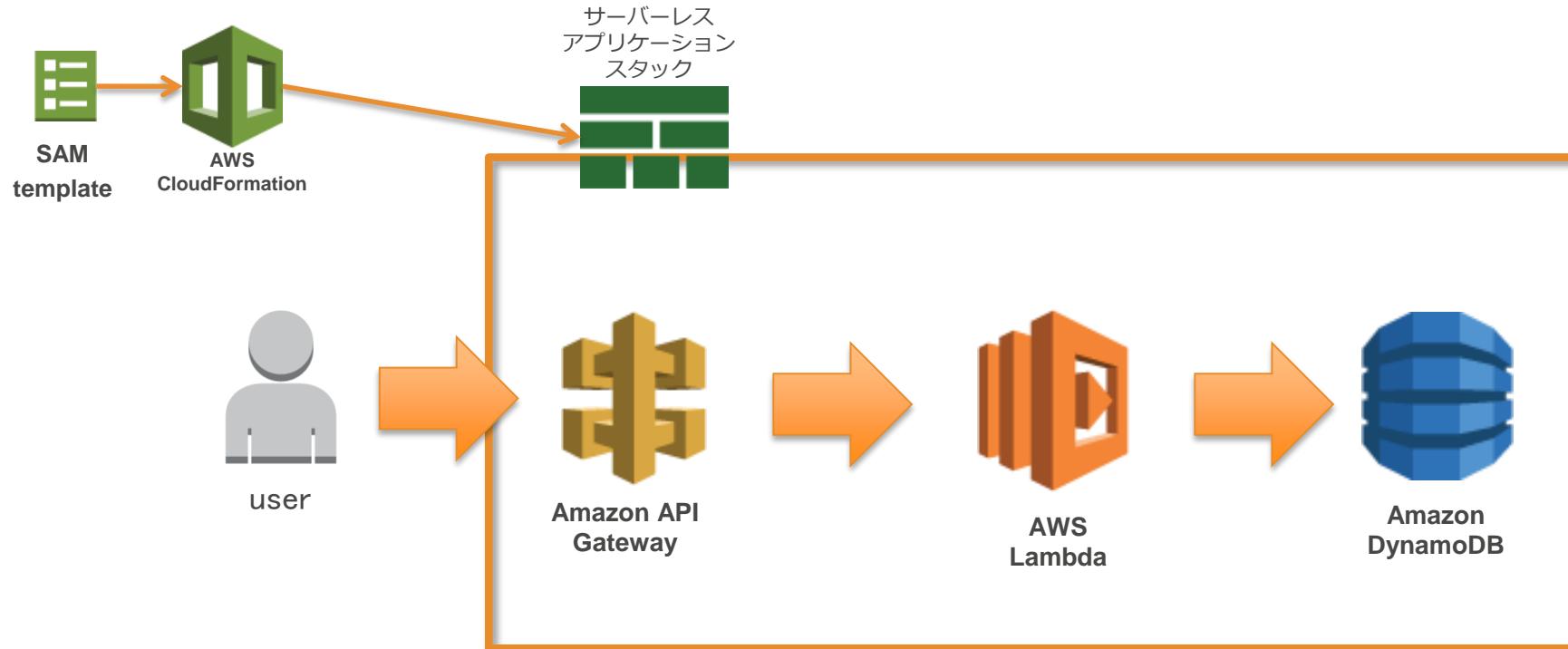
- Awslabs Githubにて公開 (<https://github.com/awslabs/serverless-application-model>)



Serverless Application Model (SAM)

New

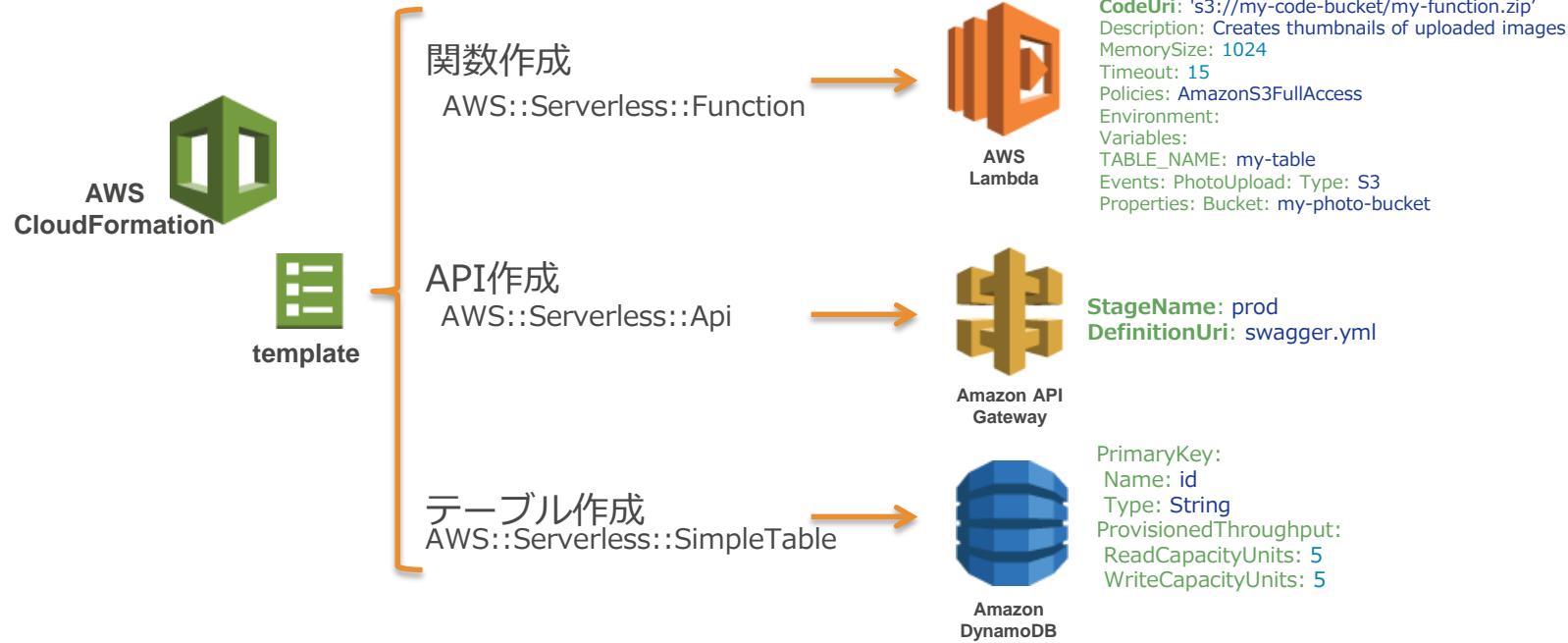
- DynamoDBにデータを格納、参照するAPIをサーバレスで簡単に一つのスタックとして構築が可能に！



Serverless Application Model (SAM)

New

SAMのTemplateの中身



CloudFormationの利用パターン

クイックスタート

- セキュリティや可用性を考慮したベストプラクティス
- マイクロソフト、SAP、トレンドマイクロなどの製品をサポート
- 15分程度でアプリケーションをAWS上に展開可能。カスタマイズも可能

サンプルコード & テンプレート

- これからCloudFormationを使い始める場合
- 独自のテンプレートを短時間で作成したい場合

自らテンプレートを作成

- JSON/YAML形式でテンプレートを記述
- CloudFormerを利用する
- CloudFormation Designerを利用する

導入の容易さ

フレキシブル

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",  
  
  "Description": "Sample",  
  
  "Parameters": {  
    "KeyName": {  
      "Description": "Sample key,  
      "Type": "String"  
    }  
  },  
  "Mappings": {  
  },  
  
  "Resources": {  
    "Ec2Instance": {  
      "Type": "AWS::EC2::Instance",  
      "Properties": {  
        "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],  
        "KeyName": { "Ref": "KeyName" },  
        "ImageId": { "Fn::FindInMap": [  
          "RegionMap", { "Ref": "AWS::Region" }, "AMI" ]}  
      }  
    },  
  
    "InstanceSecurityGroup": {  
      "Type": "AWS::EC2::SecurityGroup",  
      "Properties": {  
        "GroupDescription": "Enable SSH access via port 22",  
        "SecurityGroupIngress": [ {  
          "IpProtocol": "tcp",  
          "FromPort": "22",  
          "ToPort": "22",  
          "CidrIp": "0.0.0.0/0"  
        } ]  
      }  
    }  
  }  
}
```

テンプレートの要素

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  
  "Description" : "Valid JSON strings up to 4K",  
  
  "Parameters" : {  
    set of parameters  
  },  
  
  "Mappings" : {  
    set of mappings  
  },  
  "Conditions" : {  
    set of conditions  
  },  
  "Resources" : {  
    set of resources  
  },  
  
  "Outputs" : {  
    set of outputs  
  }  
}
```

テンプレートのバージョン

テンプレートの詳細（説明文）

CloudFormation実行時にユーザ入力を求める
パラメータを定義する
(KeyPairの名前や、DBのユーザ名など)

Hashtableのようなもの
キーに応じて値を特定出来る
(例：リージョンに応じたAMI-IDなど)

条件名と条件判断内容を登録する。この条件名は
Resourcesなどでリソース作成時に利用できる。

EC2やRDSなど、スタックを構成する
リソースを定義

スタック構築後にCloudFormationから出力させ
る値（例：DNS名やEIPの値など）

必須

Resourcesとは

- EC2やELB, RDSなど、起動するサービスを指定
- リソースごとに決められたプロパティを設定する
- CloudFormationで利用可能なリソースタイプは現在42サービス、173リソース。最新リストはWebを参照

```
"Resources": {  
    "MyInstance": {  
        "Type": "AWS::EC2::Instance",  
        "Properties": {  
            "DisableApiTermination": "FALSE",  
            "ImageId": "ami-a1b6fda0",  
            "InstanceType": "t1.micro",  
            "KeyName": { "Ref": "MyKeyName" },  
            "Monitoring": "false",  
        }  
    }  
}
```

リソースタイプの定義

リソースごとのプロパティ

Parametersとは

- スタック構築時に値を定義することができる
- データ型、デフォルト値、最大最小値など型が設定可能

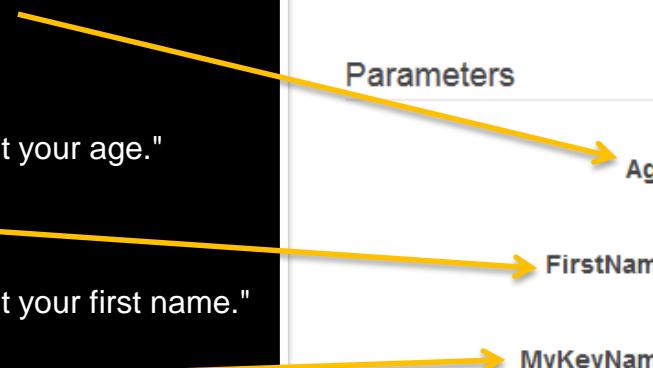
```
"Parameters": {  
    "Age":{  
        "Type":"Number",  
        "Default" : "30",  
        "MinValue": "20",  
        "MaxValue": "60",  
        "Description":"input your age."  
    },  
    "FirstName":{  
        "Type":"String",  
        "Description":"input your first name."  
    },  
    "MyKeyName": {  
        "Description" : "Input your key pair.",  
        "Type" : "String"  
    }  
},
```

Parameters

Age 32 input your age.

FirstName Masato input your first name.

MyKeyName MyKeyName Input your key pair.



Parametersのプロパティ

- 以下のプロパティが利用可能

プロパティ	内容
Type	データ型 “String” “Number” “CommaDelimitedList”
Default	デフォルト値
NoEcho	入力時に*****となる（パスワードなどに使用）
AllowedValues	入力可能値の一覧指定（例：[“true”, “false”]）
AllowedPattern	正規表現で入力可能パターンを指定（例：[a-zA-Z]*）
MaxLength	最大文字数
MinLength	最小文字数
MaxValue	最大値
MinValue	最小値
Description	プロパティの詳細説明
ConstraintDescription	入力した値がAllowedPatternやMaxLengthなどの制約に引っかかった時に表示する説明（どのような制約があるかの説明を記述）

Parametersの参照

- パラメータの値は、テンプレート中で"Ref"により参照可能
- ユーザ名やパスワードなど、ユーザ入力させたい項目に便利

Parameters

Age	<input type="text" value="32"/> input your age.
FirstName	<input type="text" value="Masato"/> input your first name
MyKeyName	<input type="text" value="MyKeyName"/> Input your key pair name

```
“Resources” : {  
    “MyServer”:{  
        “Type”:“AWS::EC2::Instance”,  
        “Properties” : {  
            “KeyName”: { “Ref” : “MyKeyName”},  
            “Tags”: [  
                { “Key”: “OwnerAge”,“Value”: { “Ref”: “Age” } },  
                { “Key”: “OwnerName”,“Value”: { “Ref”: “FirstName” } }  
            ],  
        }  
    }  
}
```

“Ref”を使うとパラメータの値を参照することができる

Function

- パラメータの参照やMapの参照などの際はFunctionを利用する
- Parameterの取得に利用した"Ref"もFunctionのひとつ
- 大別して10種類が用意されている
 - Ref → パラメータを参照する
 - Fn::Base64 → 文字列をBase64エンコードする
 - Fn::FindInMap → Mapから値を取り出す
 - Fn::GetAtt → リソースに付随する値を取得する。例) "Fn::GetAtt" : ["MyELB" , "DNSName"]
 - Fn::GetAZs → 指定したリージョンのアベイラビリティゾーンを取得する
 - Fn::Join → 文字列を結合する。例) "Fn::Join" : [":" , ["a" , "b"]] は 「a:b」 を返す
 - Fn::Select → Index値に応じた値をListから選択する。
例) { "Fn::Select" : ["1" , ["Jan" , "Feb" , "Mar" , "Apr" , "Jun"]] } は "Feb" を返す
 - 条件関数 → Fn::IfやFn::Orなどの条件分岐関連Function群
 - Fn::Sub → 指定した入力文字列を置換
 - Fn::ImportValue → 別のスタックにてエクスポートされた出力の値を取り出す。通常クロススタック参照した作成に使用（後述）

<http://docs.amazonwebservices.com/AWSCloudFormation/latest/UserGuide/index.html?intrinsic-function-reference.html>



疑似パラメータ参照 (Pseudo Parameter)

- あらかじめ定義された擬似パラメータ群で“Ref”により参照できる
- 利用可能なパラメータは下記の通り
 - AWS::Region → リージョン名を取得
 - AWS::StackId → スタックIDを取得
 - AWS::StackName → スタック名を取得
 - AWS::AccountId → AWSアカウントIDを取得
 - AWS::NotificationARNs → notification Amazon Resource Names(ARNs)を取得
 - AWS::NoValue → 指定されたプロパティを無視するようCloudFormationに伝える例)

```
"DBSnapshotIdentifier" : {  
    "Fn::If" : [  
        "UseDBSnapshot",  
        {"Ref" : "DBSnapshotName"},  
        {"Ref" : "AWS::NoValue"}  
    ]  
}
```

- **UseDBSnapshotが“True”的とき**
「DBSnapshotIdentifier」としてDBSnapshotNameの値を使う
- **UseDBSnapshotが“False”的とき**
CloudFormationはプロパティ「DBSnapshotIdentifier」が定義されていないものとして動作する (=DBSnapshotIdentifierを無視する)

```
"Resources" : {  
    "MyServer":{  
        "Type": "AWS::EC2::Instance",  
        "Properties" : {  
            "KeyName": { "Ref" : "AWS::StackName"},  
            "Tags" : { "Ref" : "AWS::Region"}  
        }  
    }  
}
```

Mappingsとは

- キーと値のマッピングテーブルを管理できる
- リージョンやユーザ入力パラメータによって、値が変わるものに利用
- Mappingsを利用してすることでテンプレートの再利用性が向上

```
"Mappings" : {  
    "RegionTable" : {  
        "us-east-1" : {  
            "AMI" : "ami-8c1fce5",  
            "KeyPair" : "myKey-east"  
        },  
        "us-west-1" : {  
            "AMI" : "ami-3bc9997e",  
            "KeyPair" : "myKey-west"  
        },  
        "ap-northeast-1" : {  
            "AMI" : "ami-300ca731",  
            "KeyPair" : "myKey-japan"  
        }},
```

Mappingsにテーブルの定義を行う
(この例の場合、"RegionTable"という
名称でMappingを定義)

Regionが"us-west-1"であれば、AMI ID
は"ami-3bc9997e"でキーペアの名前
は"myKey-west"とする

Regionが"ap-northeast-1"であれば、
AMI IDは"ami-300ca731"でキーペアの
名前は"myKey-japan"とする

Mappingsの使い方

- Functionの"Find::InMap"を使って値を取得
例) "Fn::FindInMap" : ["MapName", "Key", "Value"]
- MapName,key,Valueには"Ref"が利用可能

```
"Mappings": {  
    "RegionTable": {  
        "us-east-1": {  
            "AMI": "ami-8c1fce5",  
            "KeyPair": "myKey-east"  
        },  
        "us-west-1": {  
            "AMI": "ami-3bc9997e",  
            "KeyPair": "myKey-west"  
        },  
        "ap-northeast-1": {  
            "AMI": "ami-300ca731",  
            "KeyPair": "myKey-japan"  
        }  
    }  
},
```

リージョン名はユーザ入力

```
"Parameters": {  
    "region":{  
        "Type": "String",  
        "Description": "input your region selection."  
    }  
},  
"Resources": {  
    "MyServer":{  
        "Type": "AWS::EC2::Instance",  
        "Properties": {  
            "ImageId": {  
                "Fn::FindInMap": [ "RegionTable" , { "Ref": "region" }, "AMI" ]  
            },  
        }  
    }  
},
```

入力されたリージョンに合致するAMIの値を取得する

ユーザ入力が"ap-northeast-1"の場合、最終的な戻り値は"ami-300ca731"となる

Conditionsとは

- 条件名と成立条件を列挙
- Resourcesセクションなどで、『ある条件が成立しているときのみリソースを起動』といった条件ベースの制御が可能
- 本番環境と検証環境で構成が異なる場合に便利

```
"Parameters": {  
    "EnvType": {  
        "Description": "Environment type",  
        "Default": "test",  
        "Type": "String",  
        "AllowedValues": ["production", "test"],  
        "ConstraintDescription": "must specify prod or test."  
    }  
},
```

パラメータEnvTypeの値は、『production』
『test』のいずれかをユーザが入力する

```
"Conditions": {  
    "CreateProdResources": {"Fn::Equals": [{"Ref": "EnvType"}, "production"]}  
},
```

EnvTypeパラメータの値が"production"であれば、CreateProdResources条件が成立

Conditionsとは

- Resourcesで"Condition"を指定すると、指定した条件が成立している場合に限り、そのリソースが起動されるようになる。

```
"Resources" : {  
    "EC2Instance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]}  
        }},  
    "NewVolume" : {  
        "Type" : "AWS::EC2::Volume",  
        "Condition" : "CreateProdResources",  
        "Properties" : {  
            "Size" : "100",  
            "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone" ]}  
        }},  
    "MountPoint" : {  
        "Type" : "AWS::EC2::VolumeAttachment",  
        "Condition" : "CreateProdResources",  
        "Properties" : {  
            "InstanceId" : { "Ref" : "EC2Instance" },  
            "VolumeId" : { "Ref" : "NewVolume" },  
            "Device" : "/dev/sdh"  
        }},  
}
```

CreateProdResourcesが成立しているときに限り、本番環境のみで利用するEBSボリュームをプロビジョニング

CreateProdResourcesが成立しているときに限り、本番環境のみに存在するEBSをアタッチする

Outputsとは

- スタック構築後に取得・表示したい情報の定義
- 例えばアクセスURLや、DBの通信先情報、作ったIAMユーザー名など、あとで使用するもの
- マネジメントコンソールから確認できるので、スタックに関する情報を出力すると便利（キーペア名、AZ、インスタンスIDなど）

The screenshot shows the AWS CloudFormation console interface. The top navigation bar includes tabs for Overview, Outputs (which is selected and highlighted with a red border), Resources, Events, Template, Parameters, Tags, and Stack Policy. Below the tabs is a table with three columns: Key, Value, and Description. A single row is present in the table, corresponding to the 'URL' output. The 'Value' column contains a blue hyperlink: <http://ec2-54-92-45-141.ap-northeast-1.compute.amazonaws.com>. The 'Description' column provides a tooltip: 'AWS CloudFormer Prototype URL. Use this endpoint ...'. At the bottom of the page, there is a footer with links to 'Feedback' and copyright information: '© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy' and 'Terms of Use'.

Key	Value	Description
URL	http://ec2-54-92-45-141.ap-northeast-1.compute.amazonaws.com	AWS CloudFormer Prototype URL. Use this endpoint ...

Outputsの定義

```
"Outputs" : {  
    "InstanceId" : {  
        "Description" : "InstanceId of the newly created EC2 instance",  
        "Value" : { "Ref" : "Ec2Instance" }  
    },  
    "AZ" : {  
        "Description" : "Availability Zone of the newly created EC2 instance",  
        "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone" ] }  
    },  
    "ApplicationURL" : {  
        "Description" : "URL of running web application",  
        "Value" :  
            { "Fn::Join": [ "", [ "http://", { "Fn::GetAtt": [ "Ec2Instance" , "PublicIp"] } , "/index.html"] ] }  
    }  
}
```

出力データの名称を定義

説明

値の指定

Functionを使って文字列を加工

YAMLサポート

- YAML Ain't Markup Language "マークアップ言語ではない"の略
- YAML バージョン 1.1 の仕様をサポート
- CloudFormation Designerを除く全ての機能と関数をサポート
- ドキュメントについてもJSONとYAMLを併記（一部）

<https://aws.amazon.com/jp/blogs/news/aws-cloudformation-update-yaml-cross-stack-references-simplified-substitution/>

YAMLの特徴

- 人間にフレンドリー、読みやすい書きやすい
- 括弧 {} 、 [] 不要。括弧閉じ忘れによるエラーの削減

JSON

```
{  
    "Resources" : {  
        "HelloBucket" : {  
            "Type" : "AWS::S3::Bucket",  
            "Properties" : {  
                "AccessControl" : "PublicRead",  
                "WebsiteConfiguration" : {  
                    "IndexDocument" : "index.html",  
                    "ErrorDocument" : "error.html"  
                }  
            }  
        }  
    }  
}
```

YAML

```
Resources:  
  HelloBucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      AccessControl: PublicRead  
      WebsiteConfiguration:  
        IndexDocument: index.html  
        ErrorDocument: error.html
```

YAMLの特徴

- JSONの仕様で、できなかったコメントの追加が可能に

YAML

```
AWSTemplateFormatVersion: "2010-09-09"
Description: A sample template
Resources:
  MyEC2Instance: #An inline comment
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: "ami-2f726546" #Another comment -- This is a Linux AMI
      InstanceType: t1.micro
      KeyName: testkey
      BlockDeviceMappings:
        -
          DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

YAMLの特徴

- 特にUserDataプロパティを書く際にはJSONと比べて非常に簡潔に書くことが可能

JSON

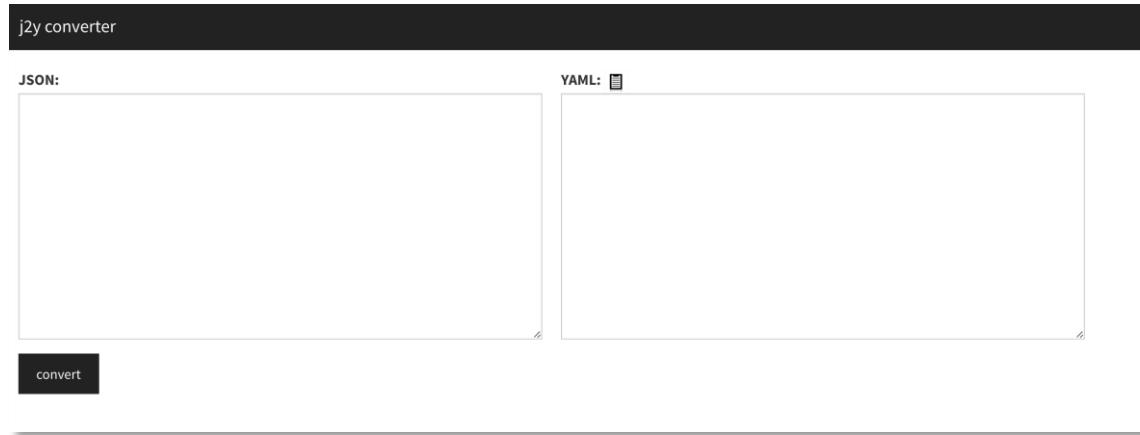
```
"UserData": { "Fn::Base64":  
  { "Fn::Join": [ "", [  
    "#!/bin/bash -xe\n",  
    "# Install the files and packages from the metadata\n",  
    "/opt/aws/bin/cfn-init -v ",  
    " --stack ", { "Ref": "AWS::StackName" },  
    " --resource WebServerInstance ",  
    " --configsets InstallAndRun ",  
    " --region ", { "Ref": "AWS::Region" }, "\n"  
  ]}  
}
```

YAML

```
UserData:  
  "Fn::Base64":  
    !Sub |  
      #!/bin/bash -xe  
      /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource LaunchConfig --configsets wordpress_install --region ${AWS::Region}
```

JSON→YAML

- 各プログラム言語にて、モジュールとしてJSONからYAMLへ変換するツールが公開されているので、それを利用する
- WEBで変換可能な3rd Partyのサービスを利用するのも便利



<http://j2y.link/>

テンプレートのまとめ

- JSONもしくはYAMLでStackの情報を記載
- リソース間の依存関係はCloudFormationが自動判別する
- Parameterを利用すれば、パラメータのユーザ入力を受け取れる
- Management Console等に値を出力したい場合はOutputsに記述する

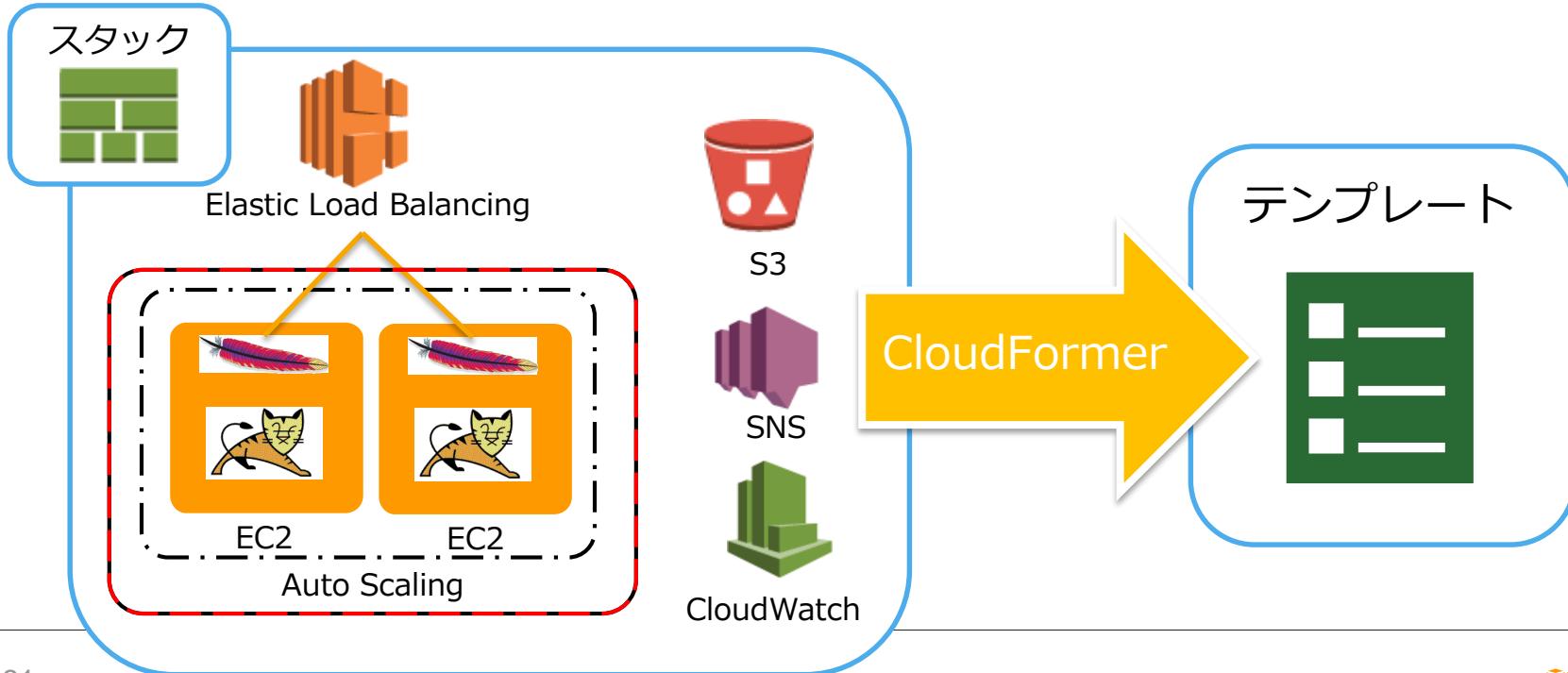
Agenda

- AWS CloudFormationの概要
- スタック、テンプレート
- テンプレート作成ツール
- AWS CloudFormationにおける運用
- まとめ



CloudFormer

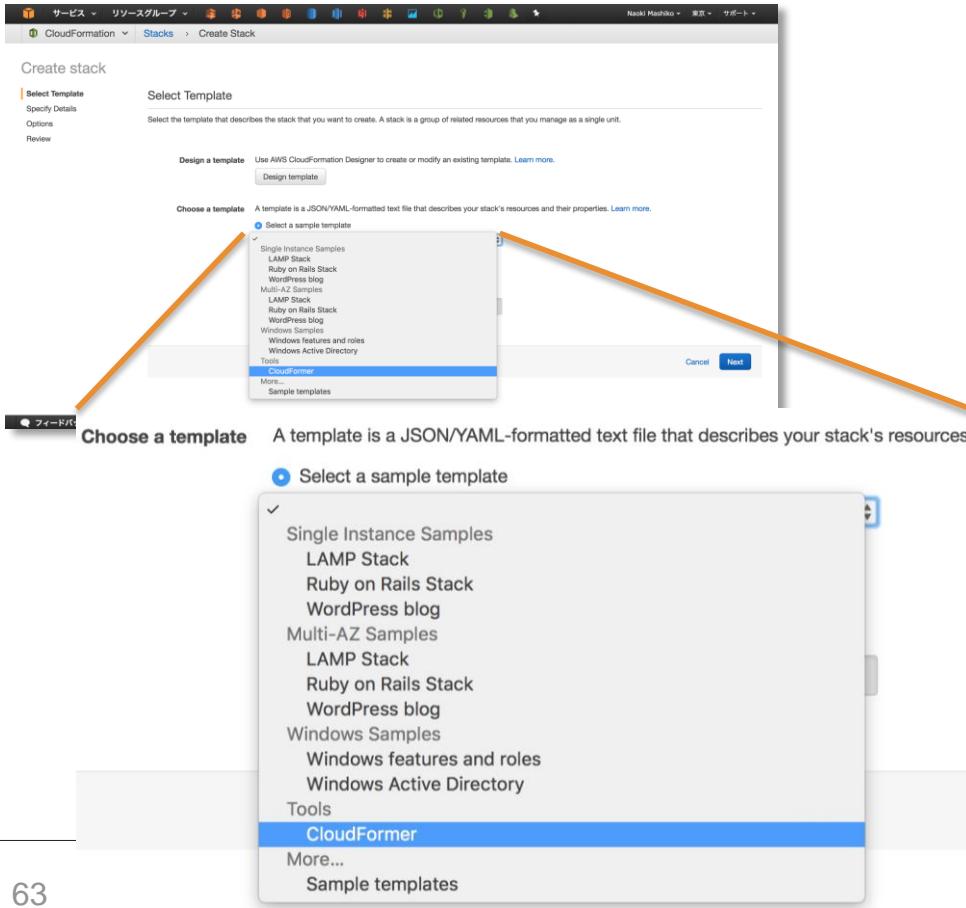
- 構築済みの環境からテンプレートを作成するツール（Webアプリ）
- テンプレート作成のベースを構築するのに有益



CloudFormerの特徴

- CloudFormerは全てのCloudFormationのリソースをサポート
- どのリソースをテンプレートに含めるか完全に制御可能
- リソースを選択すると、従属するリソースも自動選択（変更可）
 - 例えば、EC2インスタンスを選択すると、EC2インスタンスが必要とするセキュリティグループもテンプレートに含まれるようになる
- EC2インスタンスを選択すると、そのインスタンスが起動された元のAMIが指定される
 - 構築済みのイメージを利用したい場合は、生成されたテンプレートのAMI IDを確認する。意図しないものであれば修正を行う
- CloudFormerで生成されたテンプレートを手修正し、最終的なテンプレートとして利用することを推奨

CloudFormerの使い方



- 既存の手順やツールで必要なリソースを構成
- CloudFormationの「Choose Template」から、「Tools - CloudFormer」を選択
- CloudFormerがEC2のt1.microインスタンスで動作する
- CloudFormerのURLにアクセスし、画面の指示通りにテンプレート化するリソースを指定
- 生成されたテンプレートはAmazon S3に保存される
- 作業が終わったら、CloudFormerのスタックはDeleteを推奨

CloudFormerスタックの起動・アクセス

①Stackの起動が完了

Key	Value
WebsiteURL	https://ec2-52-192-96-0.ap-northeast-1.compute.amazonaws.com

②クリック

Welcome to the [AWS CloudFormation](#) template creation utility. This utility helps you to create a CloudFormation template from the AWS resources currently running in your account using a few simple steps. While the created template is complete and can be used to launch an AWS CloudFormation stack, it is a starting point for further customization. You should consider at least the following:

- Add Parameters to enable stacks to be customized at launch time.
- Add Mappings to allow the template to be customized to the specific environment.
- Replace static values with "Ref" and "Fn::GetAtt" functions to flow property data between resources where the value of one property is dependent on the value of a property from a different resource.
- Remove any static IP addresses, availability zones and other environmental properties to create more generalized configurations.
- Use CloudFormation metadata and on-host helper scripts to deploy files, packages and run commands on your Amazon EC2 instances.
- Customize any RDS Database, ElastiCache cluster or Redshift cluster passwords.
- Customize or add more stack outputs to list important information needed by the stack user.

Select the AWS Region [Asia Pacific \(Tokyo\)](#)

When you press "Create Template" we will analyze all of the AWS resources in your account. This may take a little time.

Create Template

For more information on how to build a template see the [AWS CloudFormation User Guide](#). You can also check out our [sample templates](#) demonstrating various template features.

By default, the account credentials will be used from the entries you typed in when AWS CloudFormer was created, however, they can be overridden by clicking [here](#).

③Create Templeateをクリック

CloudFormerによるテンプレート生成

- テンプレート化を行うリソースのリージョンを指定
- チェックを入れて指定したリソースが含まれるテンプレートができる
- 必要に応じてParameterの利用やOutputsの定義を行う

AWS CloudFormer



Compute Resources

Select the EC2 instances and Auto Scaling Groups to be included in the template. The volumes will be selected for inclusion in the template.

[Back](#) [Cancel](#) [Continue](#)

Auto Scaling Groups

No resources found in this AWS account

Amazon EC2 Instances

- i-a82d23a9
- i-acaa0a8ad
- i-5e0f135f
- i-680a1669
- i-920a1693
- i-523c2053
- i-5a39255b
- i-ee3529ef

AWS CloudFormer



Security Groups

Select the EC2 and RDS Security Groups to be included in the template. In case they have been modified.

[Back](#) [Cancel](#) [Continue](#)

Amazon EC2 Security Groups

- DEMO4
- test2-InstanceSecurityGroup-39UIMEJTGUIS
- ssh-web
- ElasticMapReduce-slave
- ffee-InstanceSecurityGroup-1VRAX2OHMRX5X



AWS CloudFormation Template

You can save the AWS CloudFormation template in an existing S3 bucket. Save button below. Alternatively, you can cut and paste the template code into a repository.

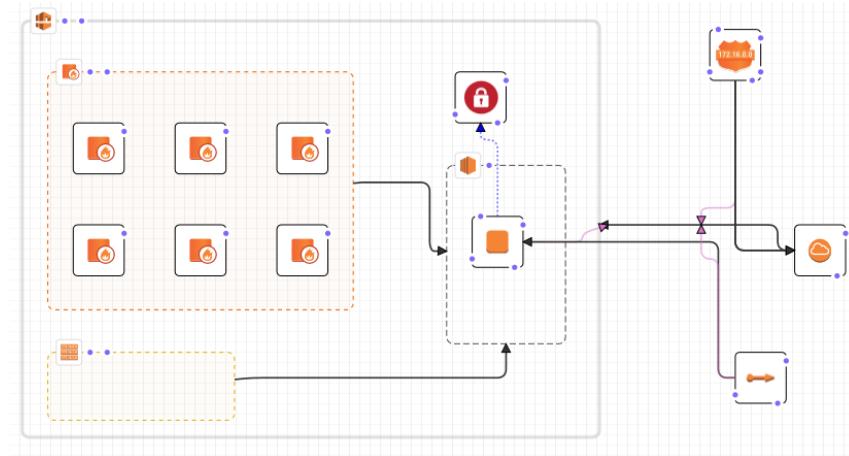
Template Name S3 Bucket

[Save Template](#) [Cancel](#)

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Resources": {  
    "eip1754123980": {  
      "Type": "AWS::EC2::EIP",  
      "Properties": {}  
    },  
    "eip1763287109": {  
      "Type": "AWS::EC2::EIP",  
      "Properties": {}  
    },  
    "eip176329542": {  
      "Type": "AWS::EC2::EIP",  
      "Properties": {}  
    }  
  }  
}
```

AWS CloudFormation Designer

- ・テンプレート内のリソースを可視化
- ・テンプレートの修正をドラッグ&ドロップで対応可能
- ・サンプルテンプレートのカスタマイズが容易

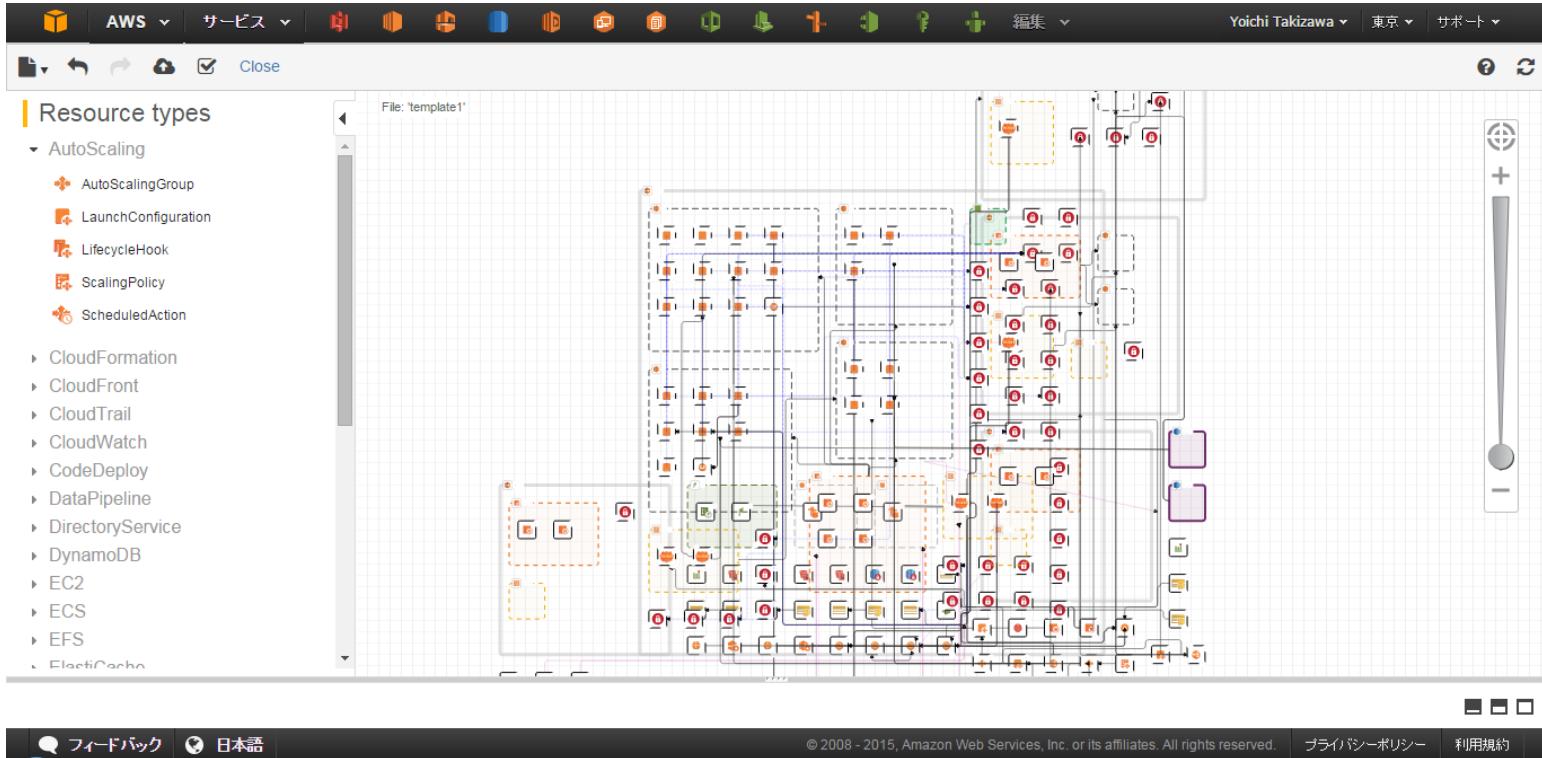


AWS CloudFormation Designer

The screenshot shows the AWS CloudFormation Designer interface. The main area is the **Canvas**, which displays a diagram of five resources connected by arrows: a lock icon (AutoScalingGroup), a hard hat icon (LaunchConfiguration), a clipboard icon (LifecycleHook), a key icon (ScalingPolicy), and a square icon (AutoScalingGroup). To the left is the **Resources** pane, which lists various resource types under the AutoScaling category. Below the canvas is the **JSONエディタ** (JSON Editor) pane, which displays the CloudFormation template JSON. The template includes a single parameter named 'Username'.

```
1: {
2:   "Parameters": {
3:     "Username": {
4:       "Description": "Username to log in to CloudFormer",
5:       "Type": "String"
6:     },
7:   }
}
```

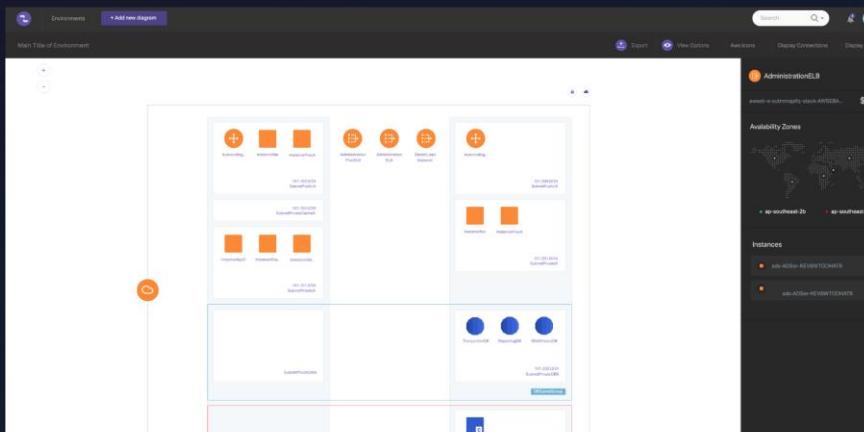
CloudFormerで作成したテンプレートを CloudFormation Designerで表示することも可能



現状の自分のAWSアカウントの状況を可視化



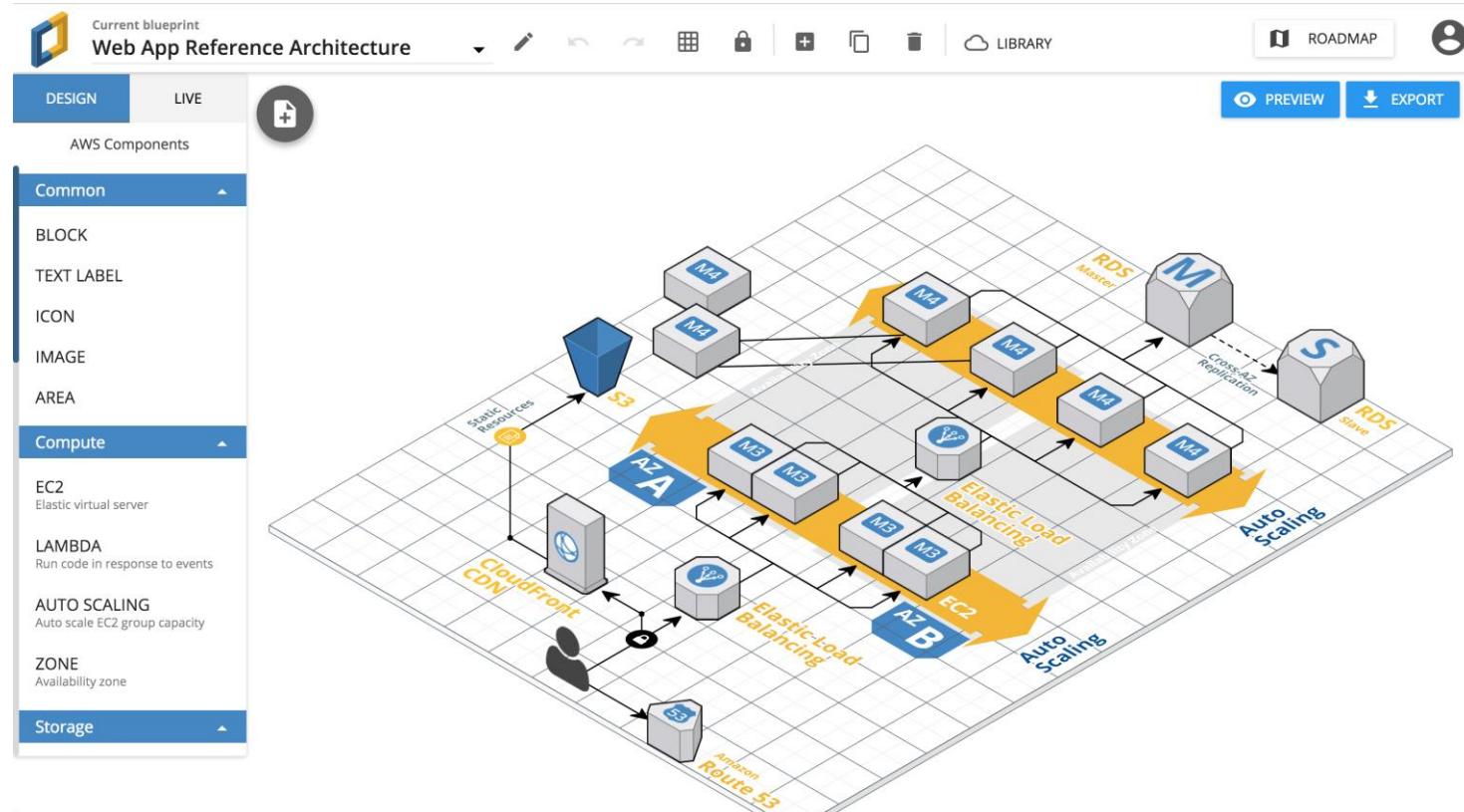
3rd party—Hava



The screenshot shows the Hava web application interface. At the top, there's a navigation bar with the Hava logo, followed by links for FEATURES, PRICING, BLOG, SIGNUP, and LOGIN. Below the navigation, a main heading reads "Your cloud architecture diagrams, automated". A subtext below it says "Free yourself from drag and drop tools. Get visibility of your existing cloud infrastructure and network topology in seconds." A prominent red button in the center says "START YOUR FREE TRIAL". The main content area displays a complex cloud architecture diagram. On the left, there's a sidebar titled "Environments" with a "+ ADD new diagram" button. The main workspace contains several interconnected components represented by icons like servers, databases, and clouds. To the right of the diagram, there's a sidebar titled "Administration" showing resource details such as "costs & estimates via AWS CloudWatch" (\$510/m), "Availability Zones" (world map with regions highlighted in green and red), and a list of "Instances" (two entries: "ami-00000000 REV1234567890" and "ami-00000001 REV1234567890").

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

3rd party – Cloudfactory



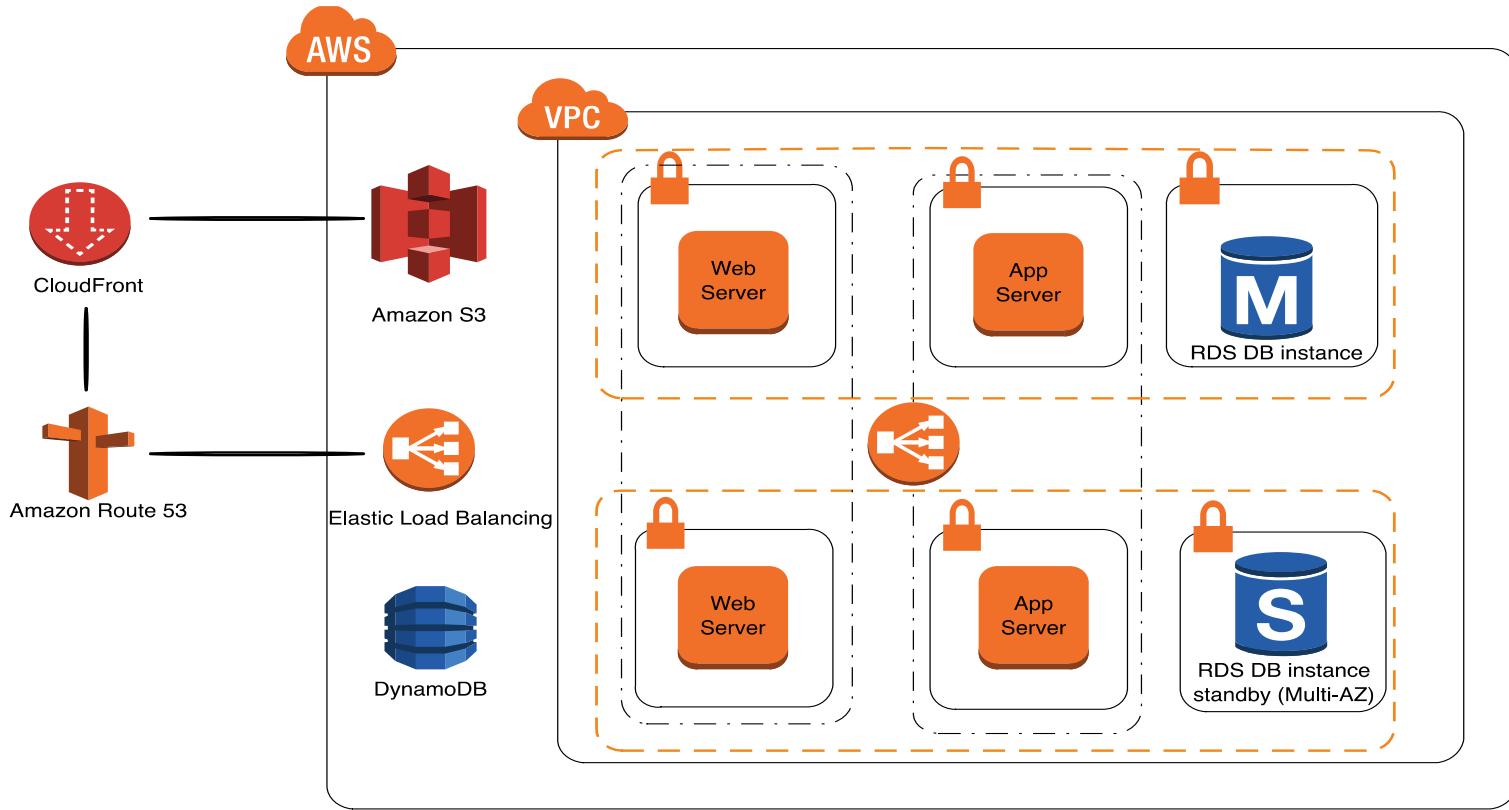
<https://cloudfactory.co/>

Agenda

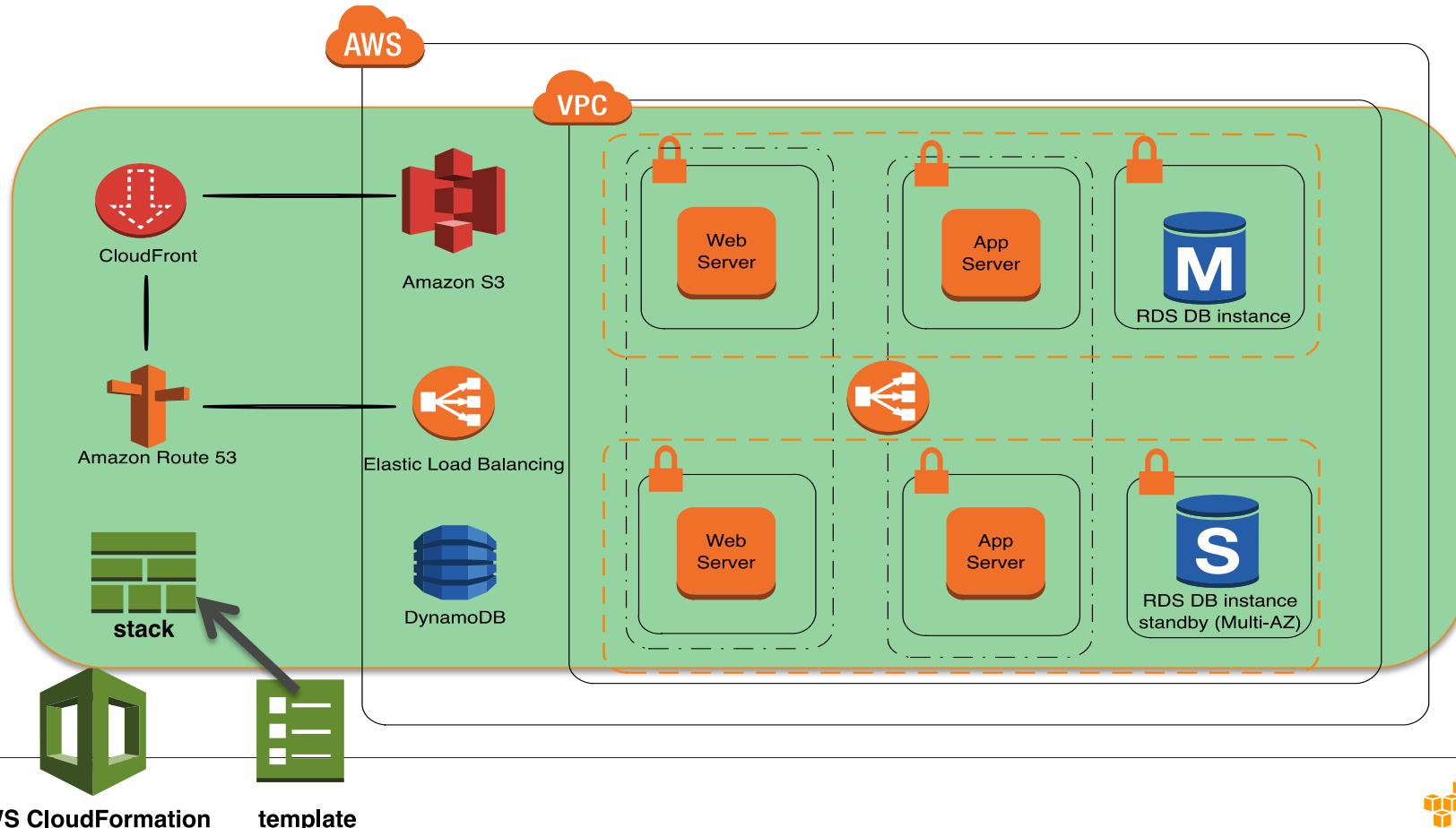
- AWS CloudFormationの概要
- スタック、テンプレート
- テンプレート作成ツール
- AWS CloudFormationにおける運用
- まとめ



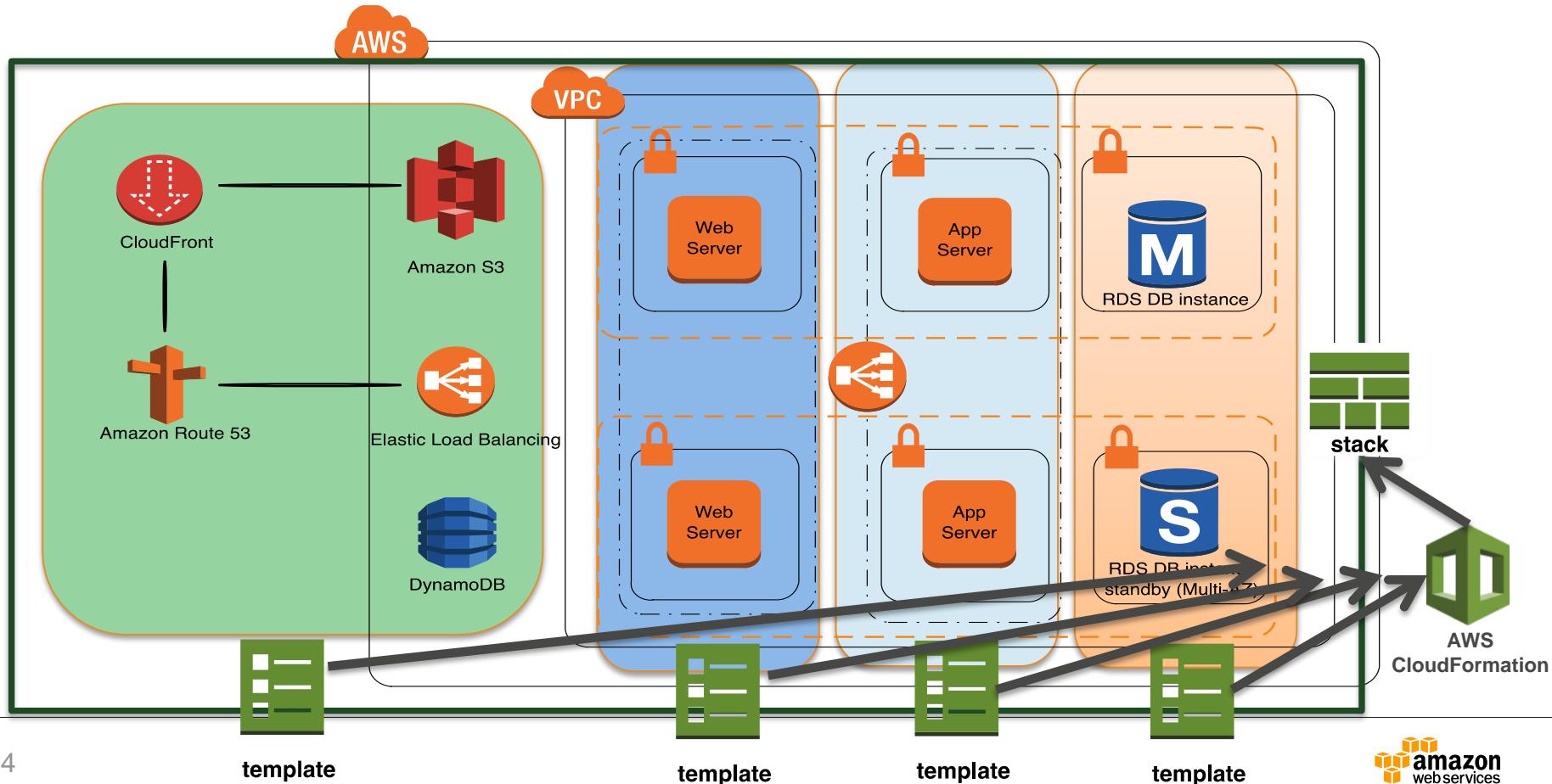
どのようにスタック、テンプレートを設計するか？



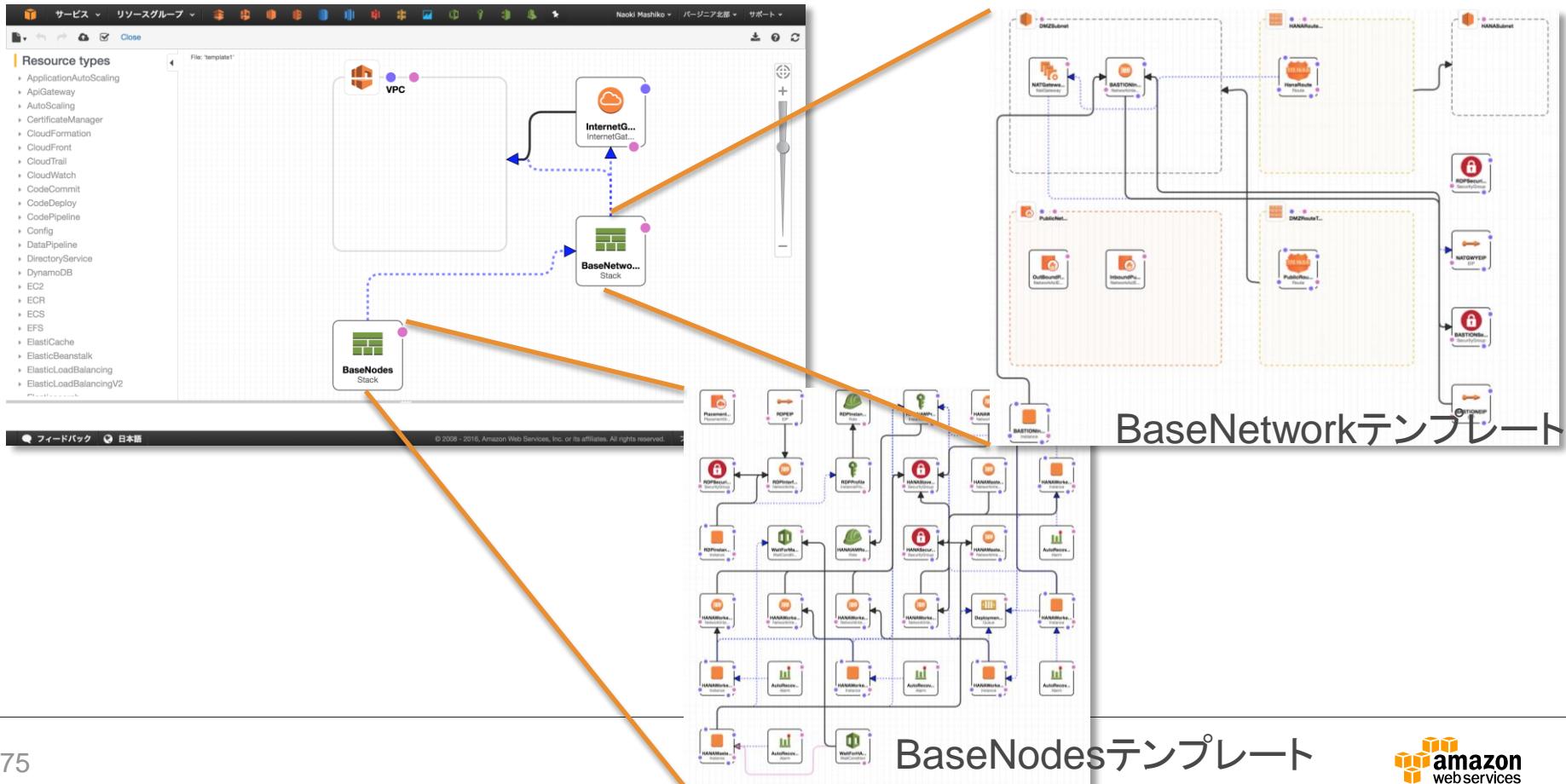
サンプルアーキテクチャ：1テンプレート、1スタック



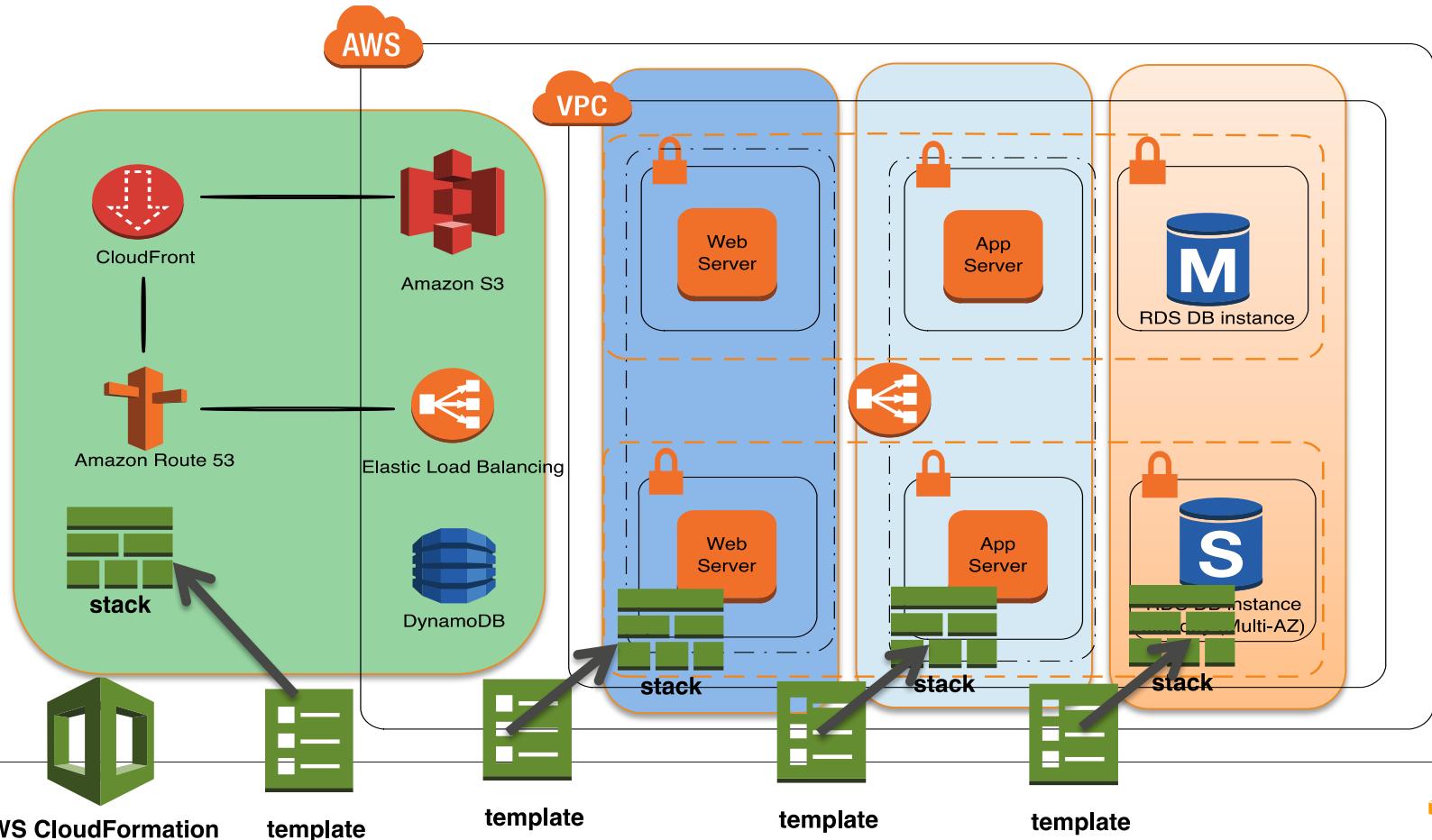
サンプルアーキテクチャ：個別テンプレート、1スタック



クイックスタートリファレンス SAP HANA



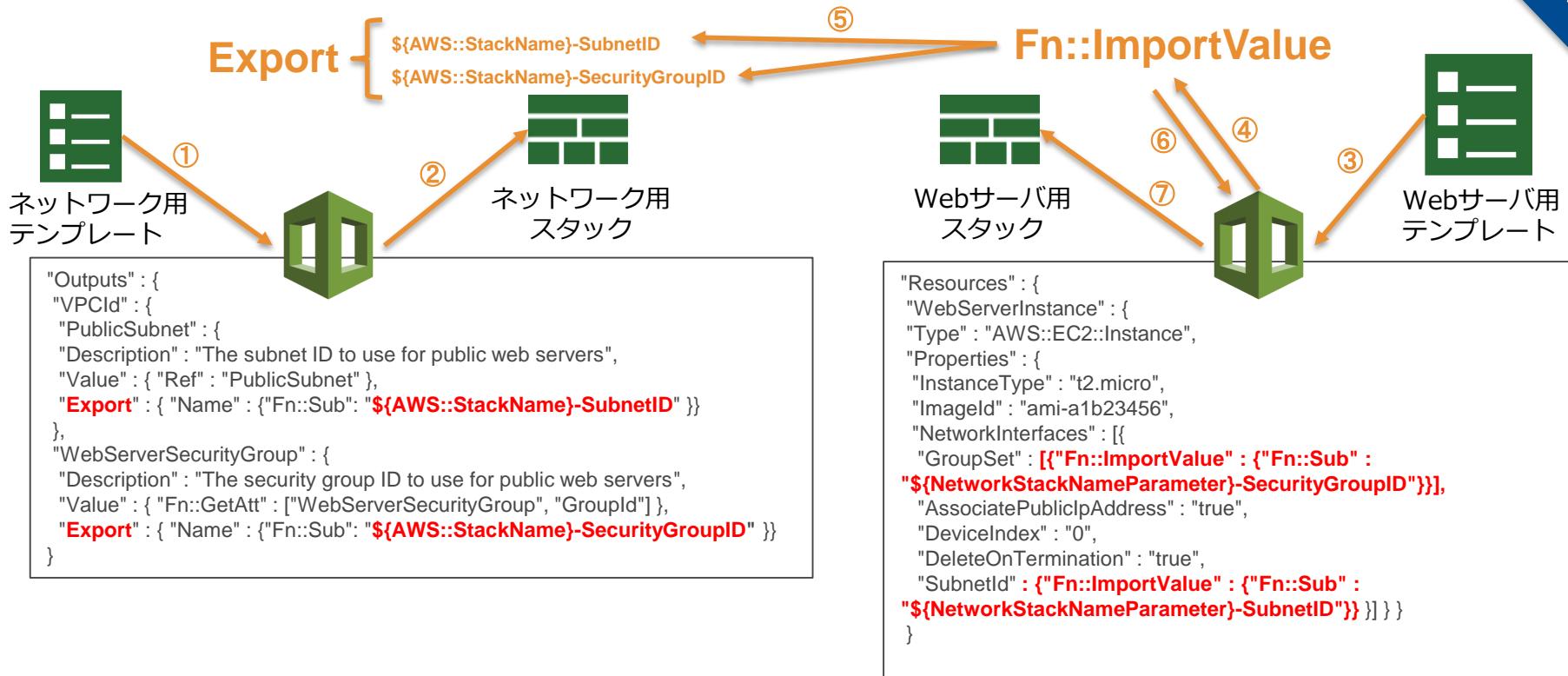
サンプルルアーキテクチャ：個別テンプレート、個別スタック



クロススタック参照

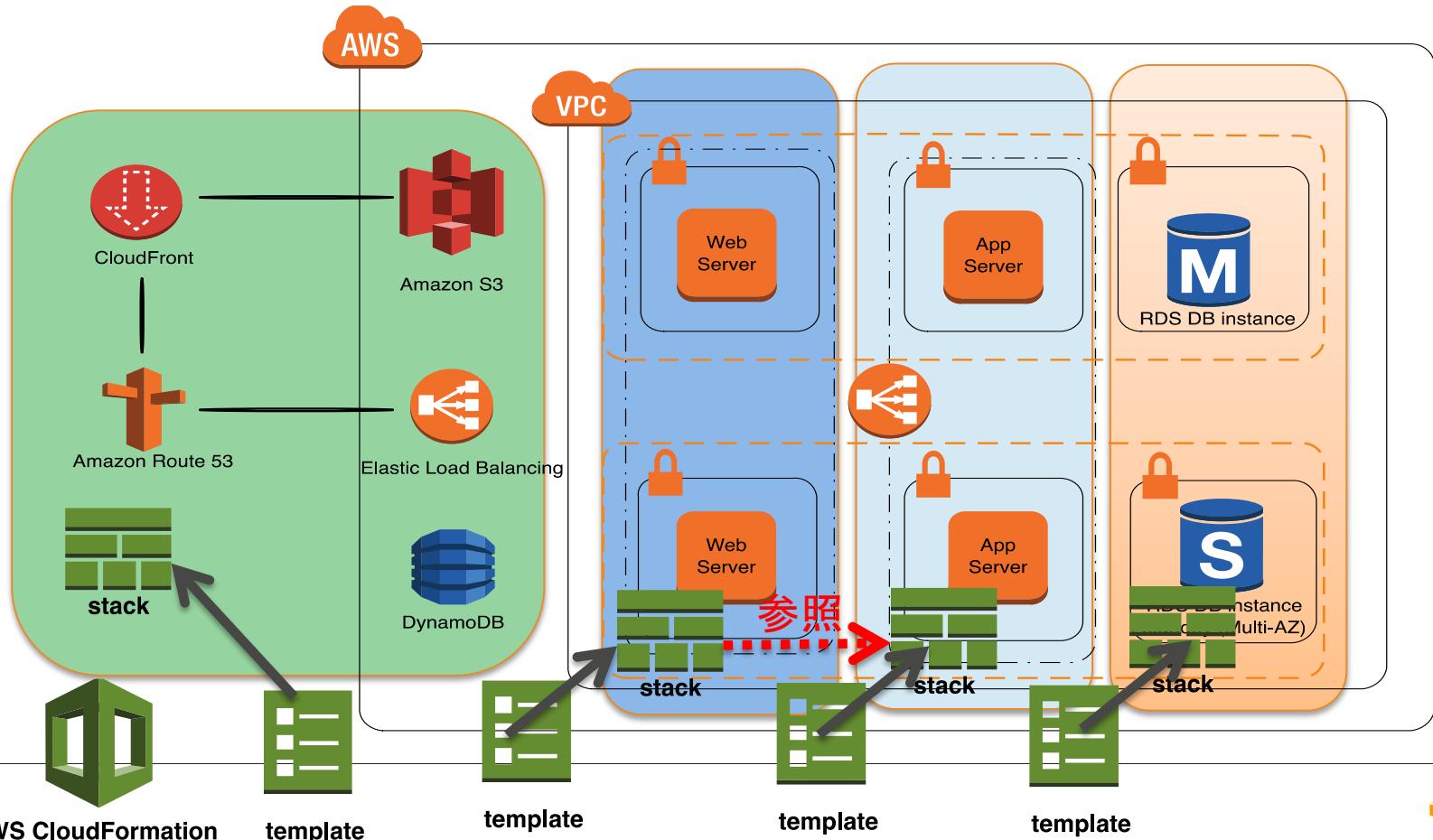
- スタックから別のスタックへリソースをエキスポート
- その値を他のスタックから参照することが可能
- これにより多層またはサービス指向アーキテクチャを実現
- 利用例：ネットワーク管理者がネットワークスタックを作成する際に、後ほど要求されるであろうセキュリティグループやサブネットの情報を予めエキスポート。ウェブ用のスタック管理者は、その情報を参照してウェブ用の新スタックを作成する等

クロススタック参照

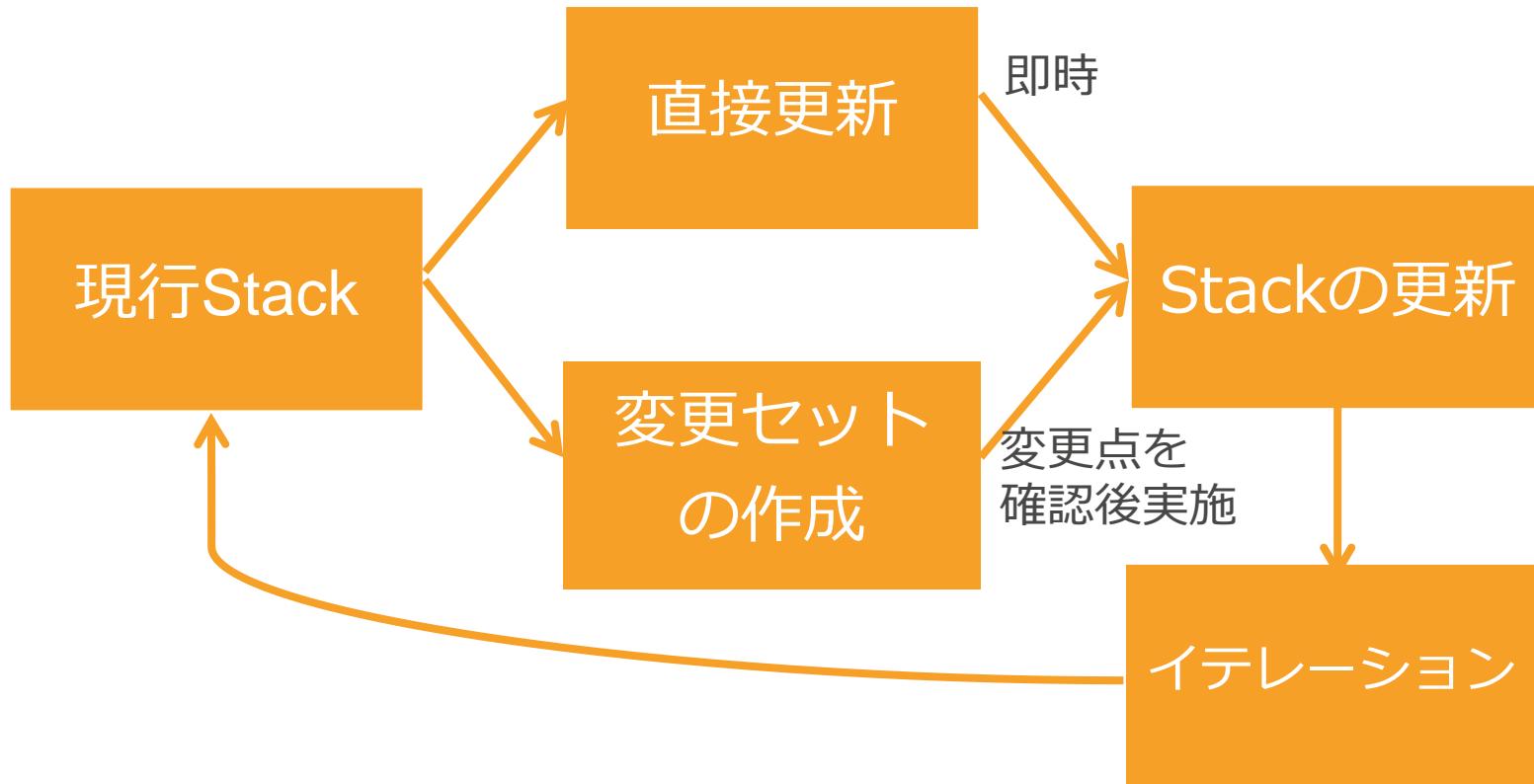


サンプルアーキテクチャ：スタック間の連携

New



Stackの更新方法



New

変更セット (Change set)

変更を要求した箇所とその変更により影響を受ける箇所を事前に確認可能

変更に伴うサービス中断の有無は各リソースの各プロパティによって変わる

プロパティ詳細 https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html

▼ Change set input

The information that was submitted. CloudFormation used this information to generate this change set.

Tags

Key	Value
Name	sample-template

変更要求した箇所

Parameters

Key	Value
InstanceType	t2.small
KeyName	demo-key
SSHLocation	10.0.0.0/16

影響内容

Amazon SNS topic ARN

No values submitted

Capabilities

No values submitted

▼ Changes

The changes CloudFormation will make if you run this change set.

Filter

Action	Logical ID	Physical ID	Resource type	Replacement
Modify	EC2Instance	i-0d74dcc7f75f36b60	AWS::EC2::Instance	True
Modify	InstanceSecurityGroup	sample-template-ec2-with-sg-InstanceSecurityGroup-1M854LDK95K5R	AWS::EC2::SecurityGroup	False

影響するリソース

※構文や差分のチェックはされるが、設定値の正しさのチェックはされないので注意が必要

New

直接更新(UPDATE)と変更セット(ChangeSet)

The screenshot shows the AWS CloudFormation console interface. It features three main sections:

- Top Left:** Shows a list of existing stacks. A red circle highlights the "Actions" dropdown menu, and an orange arrow points from the "Create Stack" button below it to this menu.
- Top Right:** A detailed view of a specific stack named "sample-template". A red circle highlights the "Actions" dropdown menu, which is open and displays options: "Create Change Set For Current Stack", "Update Stack", "Delete Stack", and "View/Edit template in Designer".
- Bottom Left:** A separate window or tab showing the "Create Stack" button highlighted with a red circle. Below it is a button for "Create Change Set For New Stack".
- Bottom Right:** Another view of the "Actions" dropdown menu for the "sample-template" stack, identical to the one in the top right section.

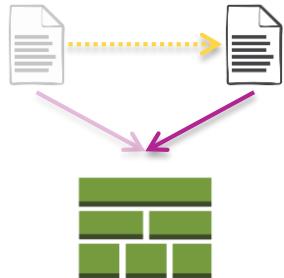
現状のスタックとの変更点を
確認して新規作成する場合

現状のスタックとの変更点を
確認して更新する場合

現状のスタックを
直接更新する場合

Stackの更新スタイル

in-place



迅速

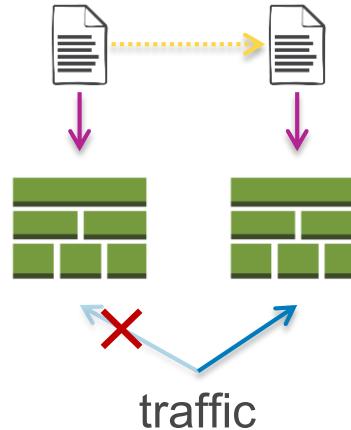
コストミニマム

簡単な状態と
データの移行

templates

stacks

blue-green



traffic

切り替え前の
テストが可能

Agenda

- AWS CloudFormationの概要
- スタック、テンプレート
- テンプレート作成ツール
- AWS CloudFormationにおける運用
- まとめ



まとめ

- 開発や本運用に必要な、互いに関連する AWS リソースのコレクションを作成しておき、そのリソースを適切な順序でプロビジョニング可能
- クイックスタートやサンプルテンプレートも充実
- AWS CloudFormation Designerで、ドラッグアンドドロップインターフェイスを使用して、テンプレートを図として視覚化し、編集することも可能
- YAMLのサポートにより、よりテンプレートを書きやすくなった
- SAMを使った迅速なサーバーレスアプリケーションの開発が可能
- 利用料金は無料。EC2やRDSなどリソースの利用料金のみ

参考資料

ドキュメント

<https://aws.amazon.com/documentation/cloudformation/>

テンプレートの分析

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-anatomy.html>

リソースタイプ

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html>

コマンドライン(CLI)

<http://docs.aws.amazon.com/cli/latest/reference/cloudformation/index.html>

サンプル

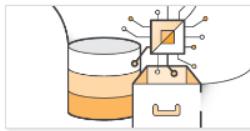
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/example-templates.html>

AWS CloudFormation よくある質問

<http://aws.amazon.com/jp/cloudformation/faqs/>

オンラインセミナー資料の配置場所

- AWS クラウドサービス活用資料集
 - <http://aws.amazon.com/jp/aws-jp-introduction/>



サービス別資料



ソリューション別資料



業種別資料



その他の資料

無料オンラインセミナー
「Black Belt Online Seminar」のサービスカット資料他、AWSのTechメンバーによる各サービスの解説資料がご覧いただけます。

無料オンラインセミナー
「Black Belt Online Seminar」のソリューションカット資料他、特定のソリューションについてのAWS活用方法がご覧いただけます。

無料オンラインセミナー
「Black Belt Online Seminar」のインダストリーカット資料他、特定の業界のユースケースがご覧いただけます。

イベントに関する資料やアップデート情報などがご覧いただけます。

- AWS Solutions Architect ブログ

- 最新の情報、セミナー中のQ&A等が掲載されています
 - <http://aws.typepad.com/sajp/>

公式Twitter/Facebook AWSの最新情報をお届けします



@awscloud_jp



もしくは
<http://on.fb.me/1vR8yWm>

最新技術情報、イベント情報、お役立ち情報、
お得なキャンペーン情報などを日々更新しています！

AWSの導入、お問い合わせのご相談

AWSクラウド導入に関するご質問、お見積り、資料請求をご希望のお客様は以下のリンクよりお気軽にご相談ください

<https://aws.amazon.com/jp/contact-us/aws-sales/>

お問い合わせ

日本担当チームへのお問い合わせ >

関連リンク

フォーラム

日本担当チームへのお問い合わせ

AWS クラウド導入に関するご質問、お見積り、資料請求をご希望のお客様は、以下のフォームよりお気軽にご相談ください。平日営業時間内に日本オフィス担当者よりご連絡させていただきます。

※ご請求金額またはアカウントに関する質問は[こちらからお問い合わせください](#)。
※Amazon.com または Kindle のサポートに問い合わせは[こちらからお問い合わせください](#)。

アスタリスク (*) は必須情報となります。

姓*

名*

※「AWS 問い合わせ」で検索してください

