



# AWS CloudFormation

## # 2 基礎編

上原 優樹 (Uehara Yuki)

Cloud Support Engineer  
2023/12

# 自己紹介

名前：上原 優樹 (Uehara Yuki)

所属：アマゾン ウェブ サービス ジャパン合同会社  
技術支援本部 クラウドサポートエンジニア

好きなAWSサービス：



AWS CloudFormation



AWS Control Tower



# 本セミナーの対象者

## 想定聴講者

- CloudFormation をこれから利用される方、概要をお知りになりたい方

## 前提

- AWS の基本的な概要や操作を理解していること
- CloudFormation の基本的な用語 (スタック、テンプレート、変更セットなど) を理解していること

※ 次の Black Belt Online Seminar で解説しています

AWS CloudFormation #1 基礎編

資料 [https://pages.awscloud.com/rs/112-TZM-766/images/AWS-Black-Belt\\_2023\\_CloudFormation-1\\_0731\\_v1.pdf](https://pages.awscloud.com/rs/112-TZM-766/images/AWS-Black-Belt_2023_CloudFormation-1_0731_v1.pdf)

動画 <https://youtu.be/4dyiPsYXG8I>

## ゴール

- CloudFormation を利用する上で必要な知識 (依存関係、動的参照など) をご理解いただくこと
- CloudFormation の利用方法 (ネストされたスタック、スタックセットなど) についてイメージを掴んでいただくこと

# アジェンダ

## 1. リソースの依存関係

- DependsOn 属性による明示的な依存関係
- 組み込み関数による暗黙的な依存関係

## 2. 動的参照

## 3. ネストされたスタック

## 4. クロスアカウント参照

## 5. StackSets



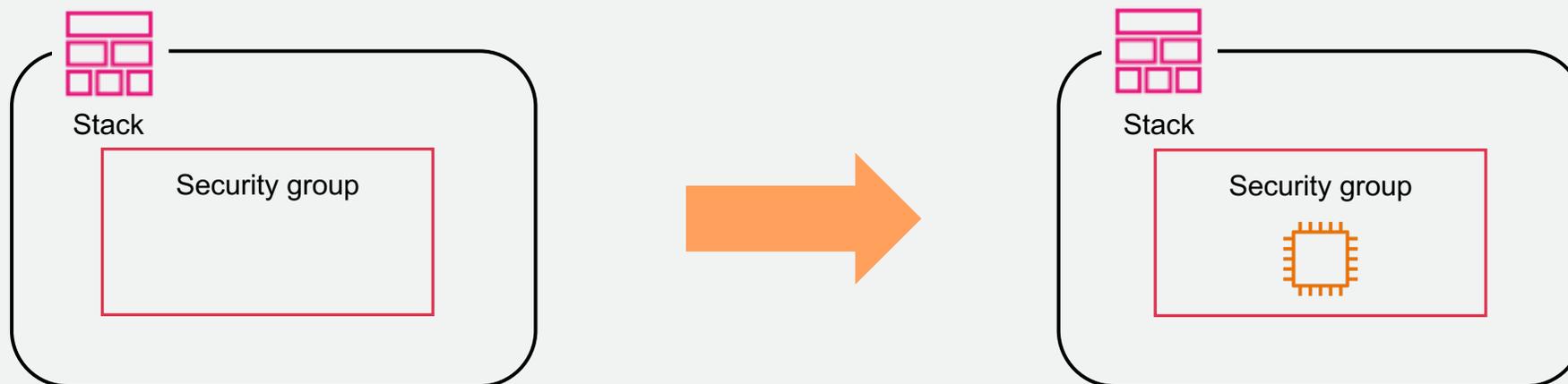
※ 本資料では CloudFormation = CFn と略記することがあります

# リソースの依存関係

# リソースの依存関係

テンプレートに記述したリソースを作成する際、リソースの構築順は、テンプレートで定義したリソース間の**依存関係**から CloudFormation が自動的に決定する。

例：EC2 インスタンスがセキュリティグループを参照するように記述することで、最初にセキュリティグループが作成され、次に EC2 インスタンスが作成される。



テンプレートで定義したリソース間に依存関係がない場合、CloudFormation はリソースの作成を**並行**して開始する。

# リソースの依存関係

## 1. DependsOn 属性による明示的な依存関係

DependsOn 属性を使用して依存関係を明示的に定義することで、リソースを決まった順序で処理することが可能。

### DependsOn 属性が必須の場合の例

- Amazon VPC ゲートウェイのアタッチメント
- Amazon ECS サービスと Auto Scaling グループ
- AWS IAM ロールポリシー

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/aws-attribute-dependson.html#gatewayattachment](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/aws-attribute-dependson.html#gatewayattachment)

## 2. 組み込み関数による暗黙的な依存関係

組み込み関数 (Fn::Ref、Fn::GetAtt、Fn::Sub) を利用し、プロパティの値として、別のリソースの属性値を参照すると、参照先のリソースが作成されてから対象のリソースを作成する。

<https://catalog.workshops.aws/cfn101/ja-JP/intermediate/templates/resource-dependencies>

# DependsOn 属性による 明示的な依存関係

# DependsOn 属性による明示的な依存関係

DependsOn 属性を使用することでリソースの**依存関係を定義**することが可能。

例：DependsOn 属性の値として Amazon S3 バケットリソースの論理 ID を使用することで、Amazon S3 バケットの作成が完了するのを待ってから、Amazon SNS トピックの作成を開始する。

```
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      Tags:
        - Key: Name
          Value: Resource-dependencies-workshop

  SNSTopic:
    Type: AWS::SNS::Topic
    DependsOn: S3Bucket
    Properties:
      Tags:
        - Key: Name
          Value: Resource-dependencies-workshop
```

論理 ID	ステータス
resource-dependencies-lab-dependson	✔ CREATE_COMPLETE
SNSTopic	✔ CREATE_COMPLETE
SNSTopic	ⓘ CREATE_IN_PROGRESS
SNSTopic	ⓘ CREATE_IN_PROGRESS
S3Bucket	✔ CREATE_COMPLETE
S3Bucket	ⓘ CREATE_IN_PROGRESS
S3Bucket	ⓘ CREATE_IN_PROGRESS
resource-dependencies-lab-dependson	ⓘ CREATE_IN_PROGRESS

# 組み込み関数による 暗黙的な依存関係

# 組み込み関数による暗黙的な依存関係 (Fn::Ref)

Fn::Ref や Fn::GetAtt などの組み込み関数を使用して他のリソースを参照することで、**暗黙的な依存関係を定義**することが可能。

例 : CloudFormation は SNSTopic リソースの作成が完了するのを待ってから、SNSTopicSubscription リソースの作成を開始する。

Resources:

SNSTopic:

Type: AWS::SNS::Topic

Properties:

Tags:

- Key: Name

Value: Resource-dependencies-workshop

AWS::SNS::Topic は !Ref による参照で Topic の ARN を返す

SNSTopicSubscription:

Type: AWS::SNS::Subscription

Properties:

Endpoint: !Ref EmailAddress

Protocol: email

TopicArn: !Ref SNSTopic

TopicArn プロパティには、サブスクライブする Topic の Amazon Resource Name (ARN) が必要

# 組み込み関数による暗黙的な依存関係 (Fn::Ref)

スタックを削除すると、CloudFormation は作成順と**逆**からリソースを削除する。

```
Resources:
  SNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      Tags:
        - Key: Name
          Value: Resource-dependencies-workshop
```

```
SNSTopicSubscription:
  Type: AWS::SNS::Subscription
  Properties:
    Endpoint: !Ref EmailAddress
    Protocol: email
    TopicArn: !Ref SNSTopic
```

論理 ID	ステータス
SNSTopic	DELETED_IN_PROGRESS
SNSTopicSubscription	DELETED_COMPLETE
SNSTopicSubscription	DELETED_IN_PROGRESS
SNSTopicSubscription	DELETED_IN_PROGRESS
resource-dependencies-with-intrinsic-functions	DELETED_IN_PROGRESS

はじめに、最後に作成された  
SNSTopicSubscription リソースを削除

# 組み込み関数による暗黙的な依存関係 (Fn::Ref)

スタックを削除すると、CloudFormation は作成順と**逆**からリソースを削除する。

```
Resources:
  SNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      Tags:
        - Key: Name
          Value: Resource-dependencies-workshop
```

```
SNSTopicSubscription:
  Type: AWS::SNS::Subscription
  Properties:
    Endpoint: !Ref EmailAddress
    Protocol: email
    TopicArn: !Ref SNSTopic
```

続いて最初に作成された SNSTopic  
リソースを削除

論理 ID	ステータス
resource-dependencies-with-intrinsic-functions	✔ DELETE_COMPLETE
SNSTopic	✔ DELETE_COMPLETE
SNSTopic	ⓘ DELETE_IN_PROGRESS
SNSTopicSubscription	✔ DELETE_COMPLETE
SNSTopicSubscription	ⓘ DELETE_IN_PROGRESS

# 組み込み関数による暗黙的な依存関係 (Fn::GetAtt)

例 : Fn::Ref 同様に参照される SecurityGroup リソースが CREATE\_COMPLETE ステータスになるのを待ってから、SecurityGroupIngress の作成が開始される。

Resources:

SecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Workshop Security Group

Tags:

- Key: Name

Value: Resource-dependencies-workshop

AWS::EC2::SecurityGroup は GroupID 属性を !GetAtt に渡して参照されると、セキュリティグループの ID を返す。

SecurityGroupIngress:

Type: AWS::EC2::SecurityGroupIngress

Properties:

GroupId: !GetAtt SecurityGroup.GroupId

IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

!GetAtt を使用することで、GroupId プロパティに SecurityGroup リソースの ID を指定可能。

# 組み込み関数による暗黙的な依存関係 (Fn::GetAtt)

スタックを削除すると、CloudFormation は作成順と**逆**からリソースを削除する。

```
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Workshop Security Group
      Tags:
        - Key: Name
          Value: Resource-dependencies-workshop
```

```
SecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !GetAtt SecurityGroup.GroupId
    IpProtocol: tcp
    FromPort: 80
    ToPort: 80
    CidrIp: 0.0.0.0/0
```

論理 ID	ステータス
SecurityGroup	DELETED_IN_PROGRESS
SecurityGroupIngress	DELETED_COMPLETE
SecurityGroupIngress	DELETED_IN_PROGRESS
sg-sample	DELETED_IN_PROGRESS

はじめに、最後に作成された SecurityGroupIngress リソースを削除。

# 組み込み関数による暗黙的な依存関係 (Fn::GetAtt)

スタックを削除すると、CloudFormation は作成順と**逆**からリソースを削除する。

```
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Workshop Security Group
      Tags:
        - Key: Name
          Value: Resource-dependencies-workshop
```

```
SecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !GetAtt SecurityGroup.GroupId
    IpProtocol: tcp
    FromPort: 80
    ToPort: 80
    CidrIp: 0.0.0.0/0
```

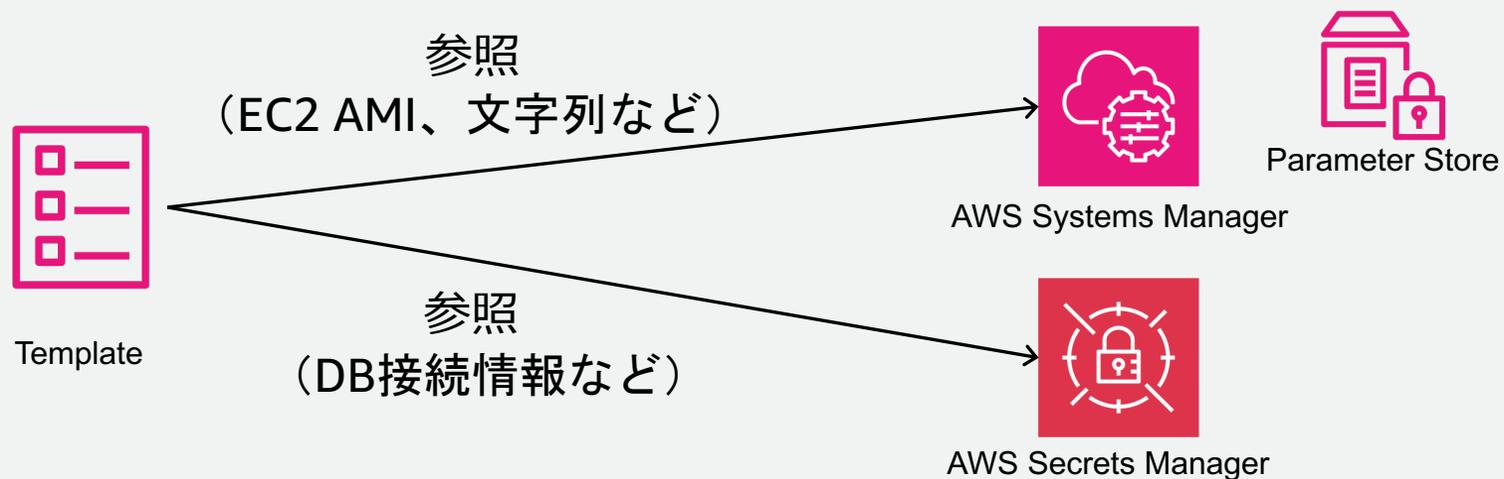
続いて最初に作成された SecurityGroup リソースを削除。

論理 ID	ステータス
sg-sample	✔ DELETE_COMPLETE
SecurityGroup	✔ DELETE_COMPLETE
SecurityGroup	ⓘ DELETE_IN_PROGRESS
SecurityGroupIngress	✔ DELETE_COMPLETE
SecurityGroupIngress	ⓘ DELETE_IN_PROGRESS

# 動的參照

# 動的参照

AWS Systems Manager (SSM) Parameter Store、AWS Secrets Manager を含む AWS サービスに保存されている外部値の参照が可能。



サービス	参照可能なデータ
AWS Systems Manager	<ul style="list-style-type: none"><li>Parameter Store に格納されている String/StringList (平文で保存されているデータ)</li><li>Parameter Store に格納されている SecureString (暗号化され保存されているデータ)</li></ul>
AWS Secrets Manager	<ul style="list-style-type: none"><li>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</li></ul>

# 動的参照 - SSM Parameter Store

## SSM パラメータ

例：スタック操作時および変更セット操作時に SSM パラメータストアに格納されている S3AccessControl パラメータバージョン 2 の値を参照し S3 バケットのアクセス制御に設定。

MyS3Bucket:

Type: 'AWS::S3::Bucket'

Properties:

AccessControl: '{{resolve:ssm:S3AccessControl:2}}'

parameter-name version

## 考慮事項

- SSM にてパラメータを更新した場合、CloudFormation にパラメータの変更を反映させるために、スタックの更新にて動的な参照を含むリソースを更新する必要がある。
- バージョンを指定しない場合、AWS CloudFormation は、スタックを作成または更新するたびに最新バージョンのパラメータを使用する。
- 現時点では、クロスアカウント SSM パラメータアクセスをサポートしていない。
- 現時点では、ドリフト検出をサポートしていない。

# 動的参照 - SSM Parameter Store

## SSM Secure String パラメータ

例：スタック操作時および変更セット操作時に SSM パラメータストアに格納されている安全な文字列であるバージョン 10 の値を IAM ユーザーのパスワードに設定。

```
MyIAMUser:  
  Type: AWS::IAM::User  
  Properties:  
    Username: 'MyUserName'  
    LoginProfile:  
      Password: '{{resolve:ssm-secure:IAMUserPassword:10}}'  
                parameter-name version
```

## 考慮事項

- セキュアな方法で参照する必要がある機密データをパラメータとして利用する際に推奨。
- Secure String パラメータの値は保存されず、API コールの結果でも返されない。
- 現時点でサポートしているリソースプロパティに対してのみ使用可能。
- 変更セットでは安全な文字列に変換された値を比較し、実際の値の比較はしない。

# 動的参照 - SSM Parameter Store

2023 年 11 月現在で動的なパラメータパターンをサポートするリソース

リソース	プロパティタイプ	プロパティ
<a href="#">AWS::DirectoryService::MicrosoftAD</a>		Password
<a href="#">AWS::DirectoryService::SimpleAD</a>		Password
<a href="#">AWS::ElastiCache::ReplicationGroup</a>		AuthToken
<a href="#">AWS::IAM::User</a>	LoginProfile	Password
<a href="#">AWS::KinesisFirehose::DeliveryStream</a>	RedshiftDestinationConfiguration	Password
<a href="#">AWS::OpsWorks::App</a>	ソース	Password
<a href="#">AWS::OpsWorks::Stack</a>	CustomCookbooksSource	Password
<a href="#">AWS::OpsWorks::Stack</a>	RdsDbInstances	DbPassword
<a href="#">AWS::RDS::DBCluster</a>		MasterUserPassword
<a href="#">AWS::RDS::DBInstance</a>		MasterUserPassword
<a href="#">AWS::Redshift::Cluster</a>		MasterUserPassword

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/dynamic-references.html#template-parameters-dynamic-patterns-resources](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/dynamic-references.html#template-parameters-dynamic-patterns-resources)



# 動的参照 - AWS Secrets Manager

## リファレンスパターン

`{{resolve:secretsmanager:secret-id:secret-string:json-key:version-stage:version-id}}`

- secret-id : 必須
  - シークレット名またはシークレット ARN。
- secret-string : 必須
  - 現在サポートされている値は SecretString のみ。
- json-key
  - 値を取得するペアのキー名。指定しない場合、シークレットテキスト全体を取得する。
- version-stage
  - シークレットのバージョンのステージングラベル。
  - version-stage を利用する場合、version-id は指定できない。
- version-id
  - シークレットのバージョンの固有識別子を指定。
  - version-id を利用する場合、version-stage は指定できない。

※ version-stage、version-id を指定しない場合、デフォルトで AWSCURRENT というバージョンが指定される。





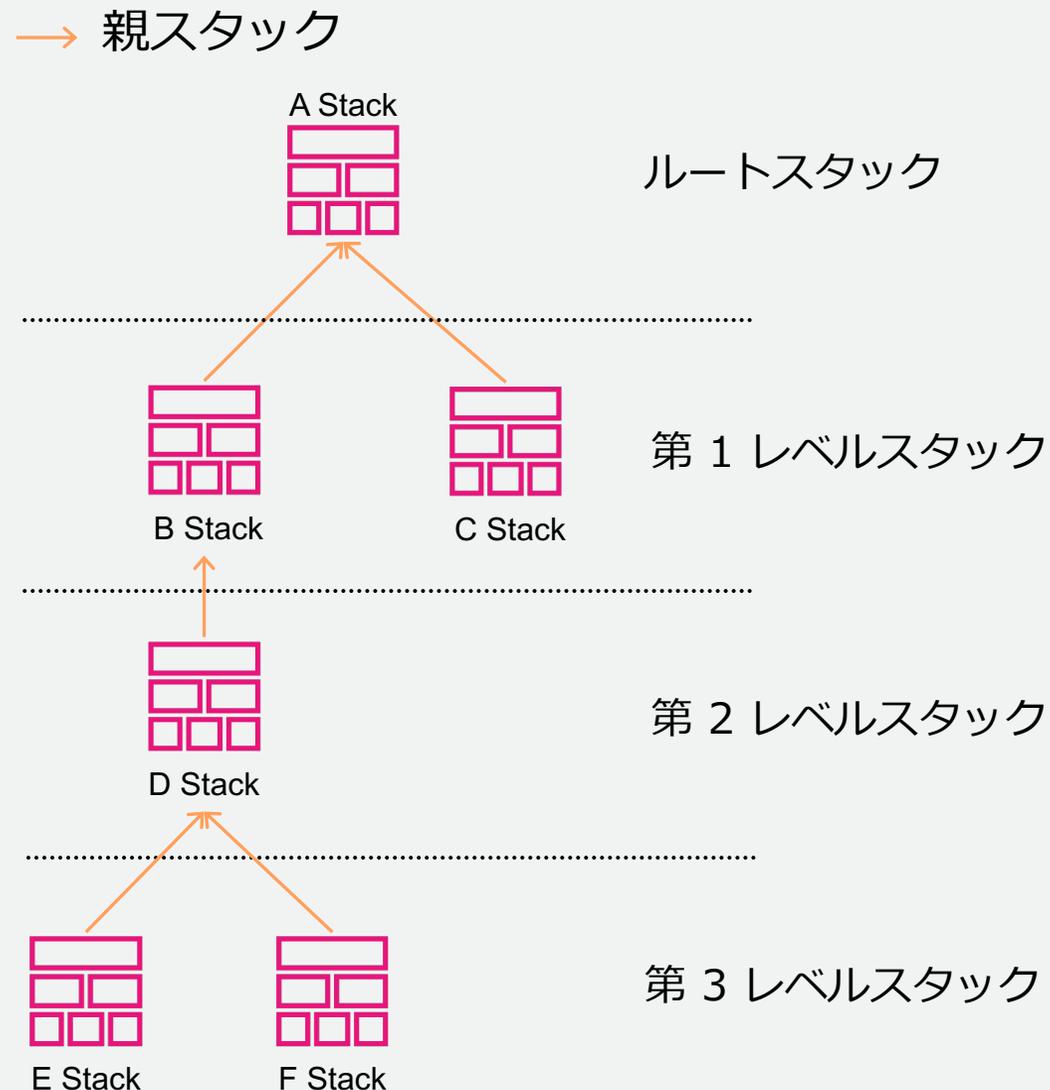
# ネストされたスタック

# ネストされたスタック

- 大きなテンプレートを小さな専用のテンプレートに分け、**参照**することが可能。
- ネストされたスタックに対しても他のスタックをネストすることが可能。
- ルートスタックは、ネストされたすべてのスタックが最終的に属する最上位スタック。
- ネストされたスタックにはそれぞれ、直接の親スタックが存在する。

## [図解]

- スタック B からみると、スタック A は親スタックであると同時にルートスタックでもある。
- スタック E からみると、スタック D が親スタック。
- スタック D からみると、スタック B が親スタック。



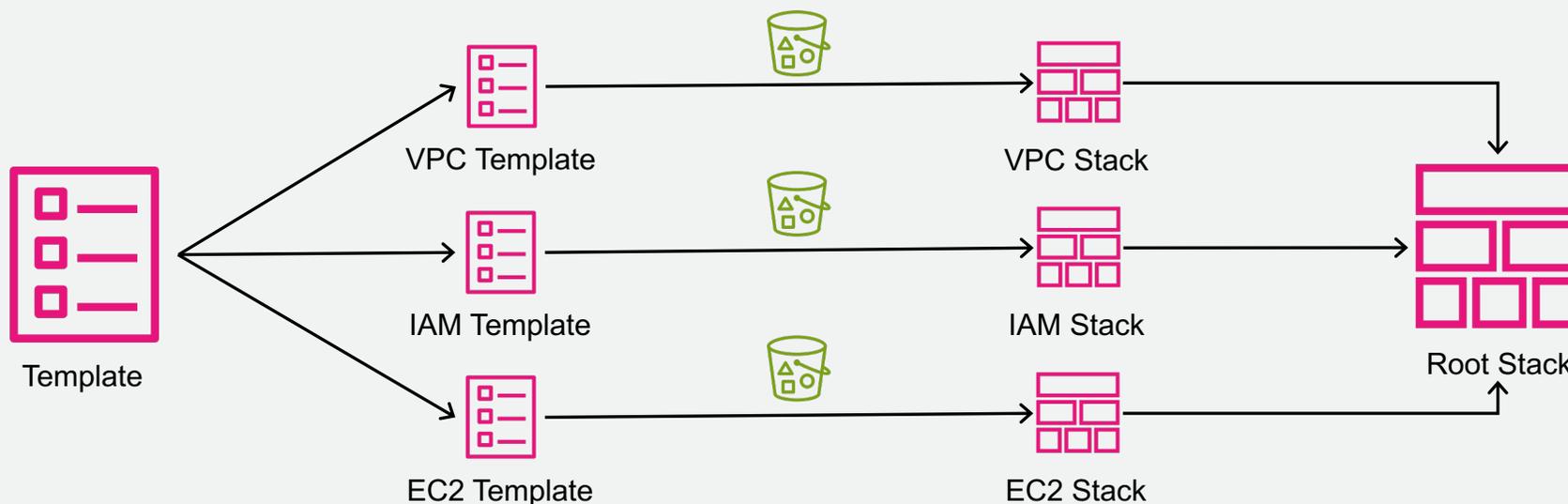
# ネストされたスタック

## ユースケース

- 複数のスタックに使用しているリソースの構成がある場合、テンプレートに同じ構成をコピーアンドペーストする代わりに、専用のテンプレートを**再利用**することが可能。
- 大きなテンプレートを小さなテンプレートに分解することでリソース制限を回避する。
- 変更セットによるリソースの変更プレビューが可能。

## 留意点

- ネストされたスタックのテンプレートを S3 バケットに予め保存する必要がある。



# ネストされたスタック（親スタック）

**AWS::CloudFormation::Stack** リソースタイプにて子スタックを定義する。

## EC2Stack:

```
Type: AWS::CloudFormation::Stack
```

### Properties:

```
TemplateURL: !Sub https://${S3BucketName}.s3.amazonaws.com/ec2.yaml
```

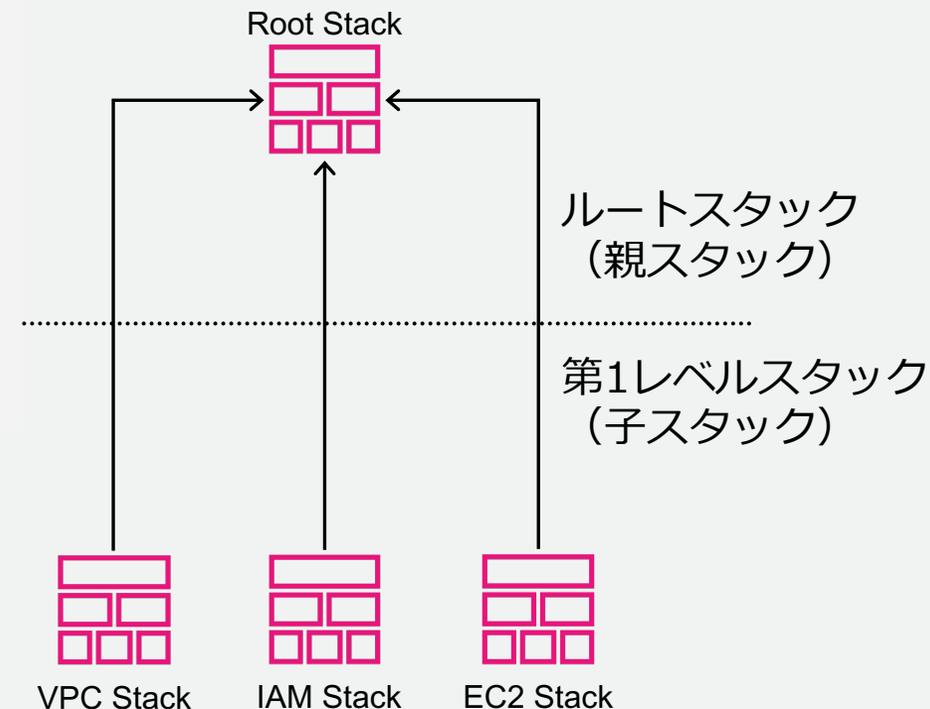
```
TimeoutInMinutes: 20
```

### Parameters:

```
EnvironmentType: !Ref EnvironmentType
```

```
VpcId: !GetAtt VpcStack.Outputs.VpcId
```

```
SubnetId: !GetAtt VpcStack.Outputs.PublicSubnet1
```



# ネストされたスタック（親スタック）

**TemplateURL** プロパティにてネストされたスタックのテンプレートを指定する。

## EC2Stack:

Type: AWS::CloudFormation::Stack

## Properties:

```
TemplateURL: !Sub https://$S3BucketName.s3.amazonaws.com/ec2.yaml
```

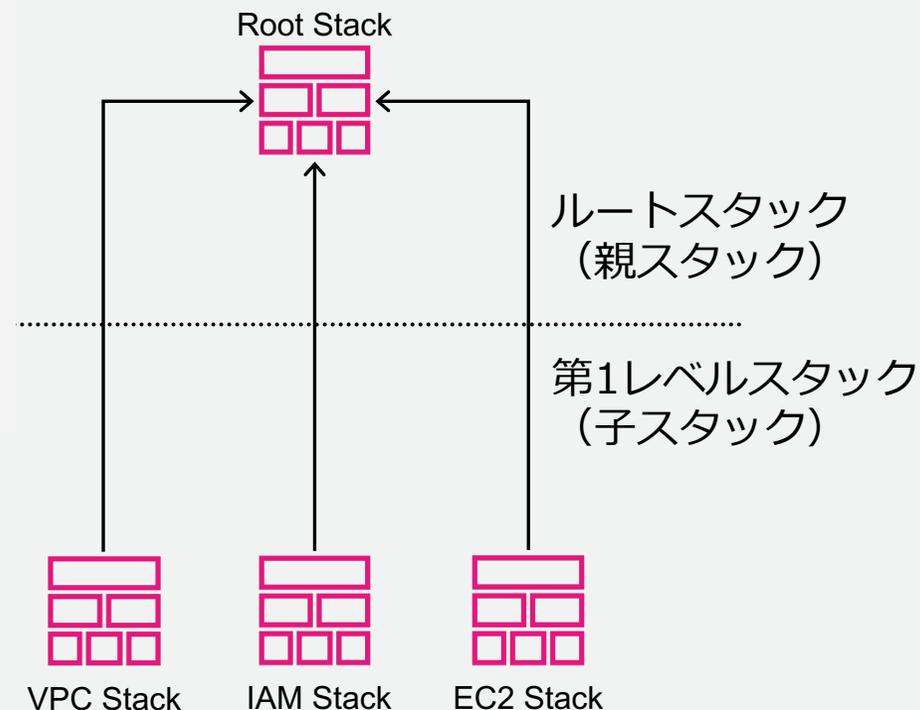
```
TimeoutInMinutes: 20
```

## Parameters:

```
EnvironmentType: !Ref EnvironmentType
```

```
VpcId: !GetAtt VpcStack.Outputs.VpcId
```

```
SubnetId: !GetAtt VpcStack.Outputs.PublicSubnet1
```



# ネストされたスタック（親スタック）

**Parameters** プロパティにてネストされたテンプレートにパラメータを渡すことが可能。

## EC2Stack:

Type: AWS::CloudFormation::Stack

### Properties:

TemplateURL: !Sub https://\$S3BucketName.s3.amazonaws.com/ec2.yaml

TimeoutInMinutes: 20

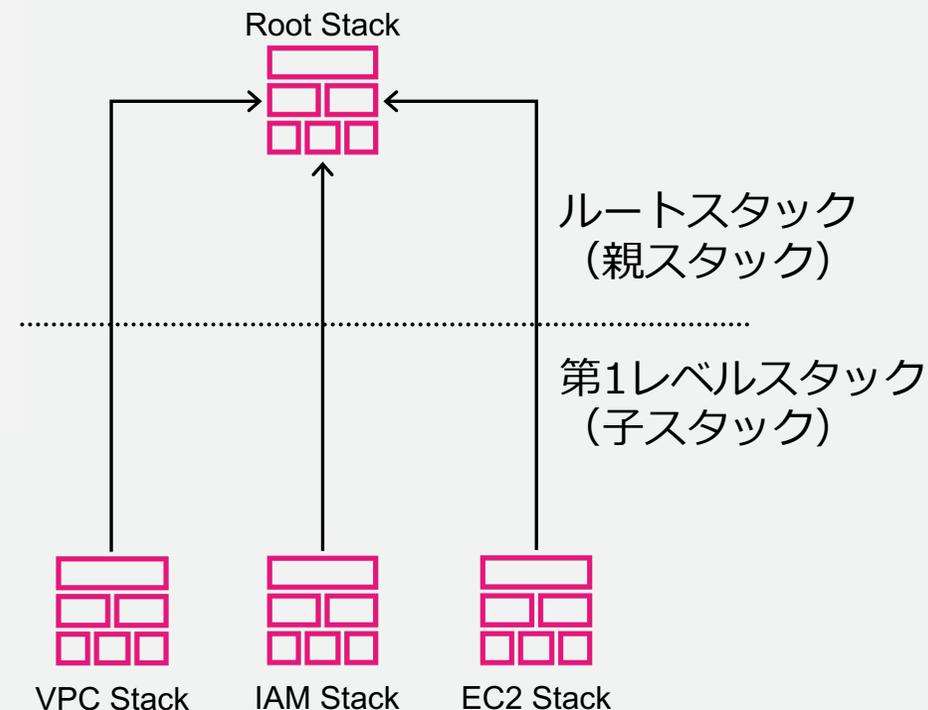
### Parameters:

EnvironmentType: !Ref EnvironmentType

VpcId: !GetAtt VpcStack.Outputs.VpcId

SubnetId: !GetAtt VpcStack.Outputs.PublicSubnet1

※ 組み込み関数 Fn::GetAtt (!GetAtt) を利用することで、他のネストされたテンプレートの Output セクションで出力された値をパラメータとして利用可能。



# クロススタック参照

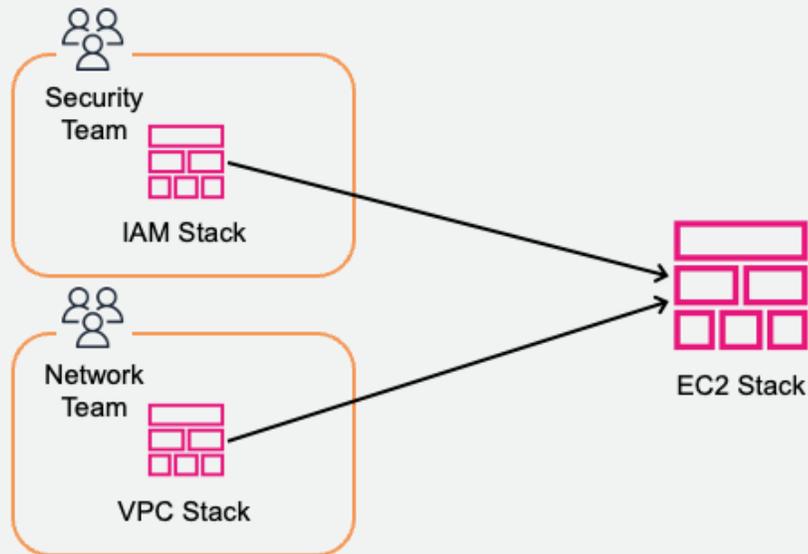
# クロススタック参照

## ユースケース

- あるスタックでデプロイしたリソースを、**他のスタックから参照したい**。
- 役割と責任を分けるために、スタックを分けて管理したい。

## 留意点

- スタックを分けすぎて管理が煩雑になる場合がある。
- 別スタックによって参照されている**出力値を変更、削除することができない**。
- スタックの出力値が参照されている場合、**当該スタックを削除することができない**。



[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/walkthrough-crossstackref.html](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/walkthrough-crossstackref.html)

# クロススタック参照

## 使い方:参照されるスタック (IAM Stack)

- Output セクションの Export フィールドに出力したい値を指定する。

### WebServerInstanceProfile:

Type: AWS::IAM::InstanceProfile

### Properties:

Path: /

### Roles:

- !Ref SSMIAMRole

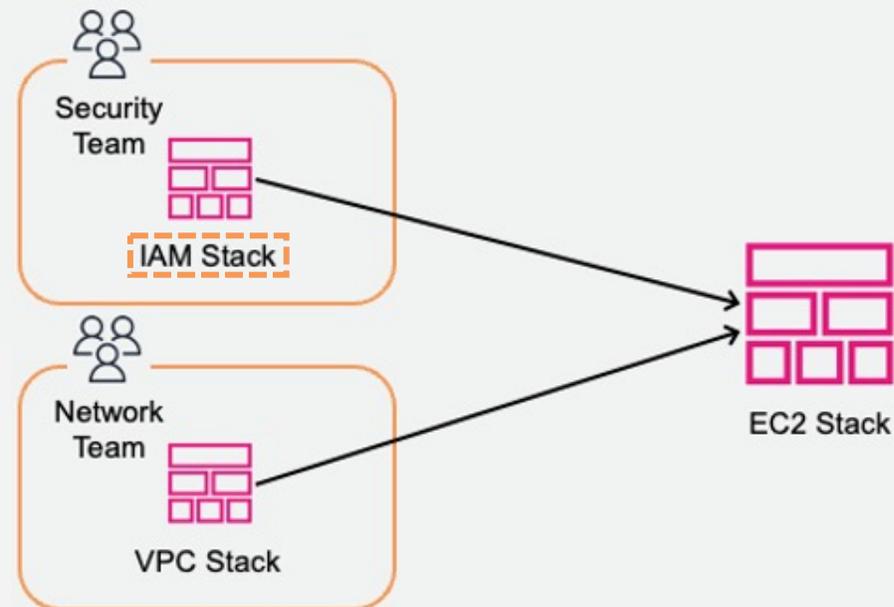
### Outputs:

#### WebServerInstanceProfile:

Value: !Ref WebServerInstanceProfile

#### Export:

Name: cfn-workshop-WebServerInstanceProfile



# クロススタック参照

## 使い方:参照するスタック (EC2 Stack)

- ・組み込み関数 Fn::ImportValue にて取得したい値を指定する。

WebServerInstance:

Type: AWS::EC2::Instance

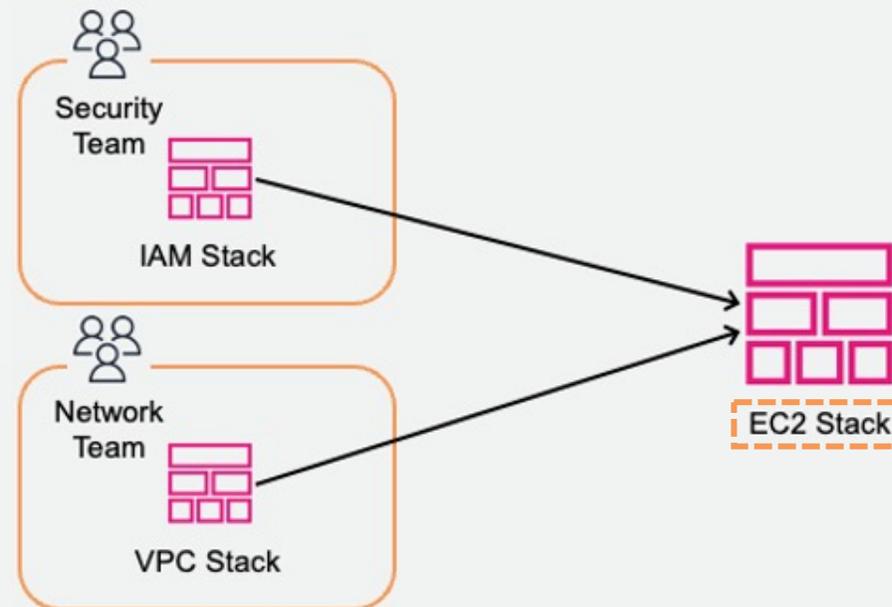
{...}

Properties:

SubnetId: !ImportValue cfn-workshop-PublicSubnet1

IamInstanceProfile: !ImportValue cfn-workshop-WebServerInstanceProfile

ImageId: !Ref AmiID



# ネストされたスタックとクロススタック参照の違い

## ネストされたスタック

- **ネストされたスタックグループ内でのみ情報共有をしたい**場合に推奨。
  - 複数の子スタックのリソースを親スタックによりすべてデプロイし、管理する。
  - 子スタックのテンプレートを S3 バケットに保存する必要がある。
  - `AWS::CloudFormation::Stack` リソースタイプを使用することで、子スタックからテンプレートを参照可能。 (\*)
  - `TemplateURL` プロパティにてネストされたスタックのテンプレートを指定。 (\*)
  - `Parameters` プロパティにてネストされたテンプレートにパラメータを渡す。 (\*)
- (\*: 親スタック)

## クロススタック参照

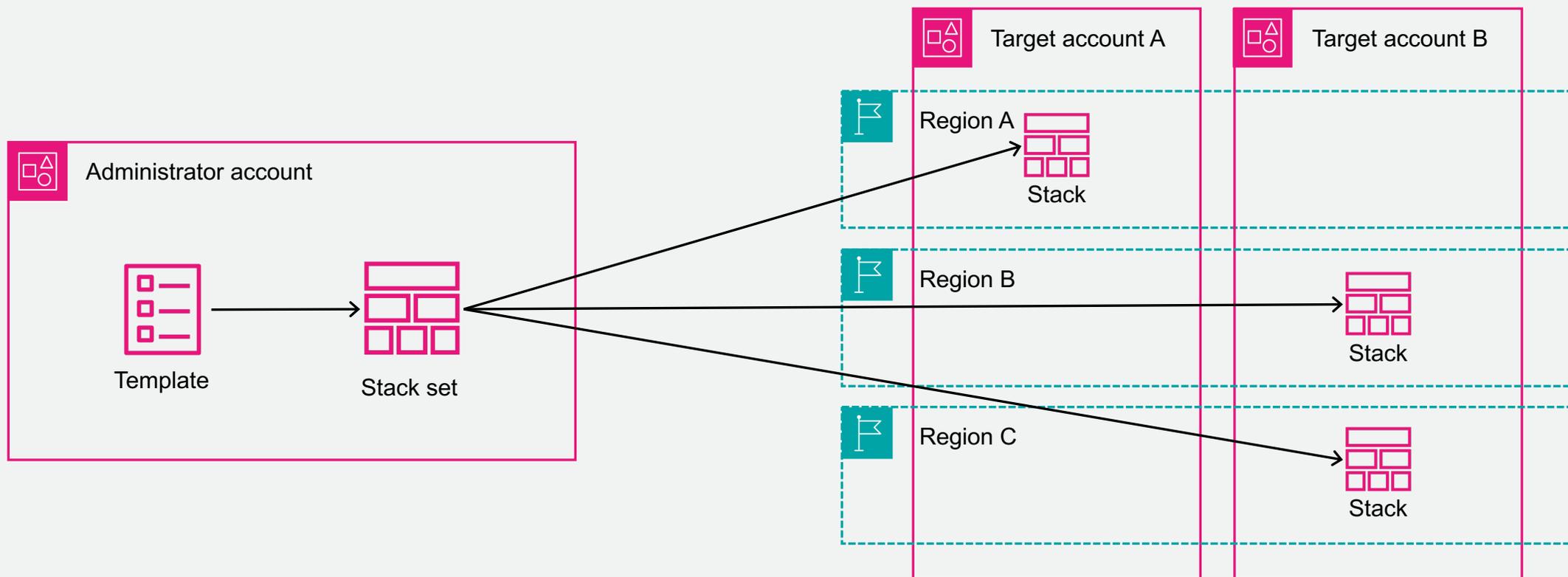
- **ネストされたスタックグループ内に限らず、他のスタックと情報を共有したい**場合に推奨。
- 他のスタックによって管理されているリソースをインポートして利用する。
- `Output` セクションの `Export` フィールドに出力したい値を指定。
- 組み込み関数 `Fn::ImportValue` にて取得したい値を指定。

# StackSets



# 概要

1 つのテンプレートを使用して複数の AWS アカウント（ターゲットアカウント）、複数のリージョンにスタックを作成、更新、削除することが可能。



# 用語

- **管理者アカウント**

- スタックセットを作成する AWS アカウント

- **ターゲットアカウント**

- スタックセットの 1 つ以上のスタックを作成、更新、削除する AWS アカウント

- **スタックセット**

- スタックの作成に使用するテンプレートおよびパラメータ、スタックを作成するターゲットアカウント、デプロイするリージョンなどの定義

- **スタックインスタンス**

- ターゲットアカウントのスタックへのリファレンス
- スタックインスタンスはスタックなしで存在可能  
(何らかの理由によりスタックが作成されていない場合、スタックインスタンスに失敗理由が表示される)
- スタックインスタンスは 1 つのスタックセットにのみ関連付けられる

# オペレーション

- **アクセス許可モデルの選択と設定（初回のみ）**

- スタックセットはセルフマネージド型またはサービスマネージド型のアクセス許可が必要。

- **スタックセットの作成**

- スタックの作成に使用するテンプレート、ターゲットアカウント、デプロイする AWS リージョンの指定を行い作成。

- **スタックセットの更新**

- スタックセットを更新すると、スタックセットのスタックに変更内容がプッシュされる。

- **スタックインスタンスのパラメータの上書き**

- アカウントおよびリージョン別にスタックインスタンスのパラメータ値を上書き可能。

- **スタックの削除**

- 指定したリージョン内の指定したターゲットアカウントから、スタックを削除する。  
(スタックの削除に併せて対象のスタックインスタンスも削除される。)

- **スタックセットの削除**

- スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。



# アクセス許可モデルの選択と設定 (初回のみ)

## セルフマネージド

 Administrator account



AWS CloudFormation StackSet  
AdministrationRole

 Target account



AWS CloudFormation StackSet  
ExecutionRole

管理アカウントとターゲットアカウントの双方に指定された名称・内容で IAM ロールを作成する必要がある。  
IAM ロール作成用 CloudFormation テンプレート配布されている。

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/stacksets-prereqs-self-managed.html](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/stacksets-prereqs-self-managed.html)

## サービスマネージド

[AWS Organizations](#) > [サービス](#) > CloudFormation StackSets

### CloudFormation StackSets

[コンソールに移動する](#)

AWS CloudFormation の拡張機能で、1 回の操作で複数のアカウントとリージョンにまたがるスタックを作成、更新、または削除できるようにします。 [詳細はこちら](#)

#### 信頼されたアクセス

[信頼されたアクセスを無効にする](#)

ステータス  
信頼されたアクセスが有効

信頼されたアクセスを有効にすると、CloudFormation StackSets が組織内の信頼されたサービスとして指定されます。信頼されたサービスは、組織の構造をクエリし、組織のアカウントにサービスにリンクされたロールを作成できます。サービスにリンクされたロールにより、信頼されたサービスは、信頼されたサービスのドキュメントに記載されているタスクを実行できます。信頼されたサービスは組織への変更についてに通知を受け、これらの通知に応じて追加のタスクを実行できます。 [詳細はこちら](#)

AWS Organizations で StackSets の信頼されたアクセスを有効化することで、管理アカウントとターゲットアカウントの双方に必要な IAM ロールが自動的に作成される。

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/stacksets-orgs-activate-trusted-access.html](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/stacksets-orgs-activate-trusted-access.html)



# オペレーション

- **アクセス許可モデルの選択と設定 (初回のみ)**
  - スタックセットはセルフマネージド型またはサービスマネージド型のアクセス許可が必要。
- **スタックセットの作成**
  - スタックの作成に使用するテンプレート、ターゲットアカウント、デプロイする AWS リージョンの指定を行い作成。
- **スタックセットの更新**
  - スタックセットを更新すると、スタックセットのスタックに変更内容がプッシュされる。
- **スタックインスタンスのパラメータの上書き**
  - アカウントおよびリージョン別にスタックインスタンスのパラメータ値を上書き可能。
- **スタックの削除**
  - 指定したリージョン内の指定したターゲットアカウントから、スタックを削除する。  
(スタックの削除に併せて対象のスタックインスタンスも削除される。)
- **スタックセットの削除**
  - スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。

# スタックセットの作成 1/2

## 1. テンプレートの選択

アクセス許可を指定し、自身で用意したテンプレートかサンプルテンプレートを指定。

### アクセス許可

IAM ロールを選択して、CloudFormation がターゲットアカウントを管理する方法を明示的に定義します。 [詳細はこちら](#)

サービスマネージドアクセス許可  
StackSets は、AWS Organizations が管理するターゲットアカウントにデプロイするために必要なアクセス許可を自動的に設定します。このオプションを使用すると、組織内のアカウントへの自動デプロイを有効にできます。

セルフサービスのアクセス許可  
ターゲットアカウントにデプロイするために必要な IAM ロールを作成します。ロールを選択しない場合、CloudFormation は現在のユーザーの認証情報に基づくアクセス許可を使用します。

### 前提条件 - テンプレートの準備

テンプレートの準備  
各スタックはテンプレートに基づきます。テンプレートとは、スタックを含む AWS リソースに関する設定情報を含む JSON または YAML ファイルです。

テンプレートの準備完了

サンプルテンプレートを使用

### テンプレートの指定

テンプレートは、スタックのリソースおよびプロパティを表す JSON または YAML ファイルです。

テンプレートソース  
テンプレートを選択すると、保存先となる Amazon S3 URL が生成されます。

Amazon S3 URL

テンプレートファイルのアップロード

Amazon S3 URL  
  
Amazon S3 テンプレートの URL

S3 URL: URL を指定すると生成されます。 デザイナーで表示

## 2. スタックセットの詳細を指定

スタックセット名称や説明、テンプレートのパラメータを指定。

### StackSet 名

StackSet 名  
  
小文字、大文字、数字、ダッシュを含める必要があります。文字で始まる必要があります。

### StackSet の説明

説明を使用して、スタックセットの目的やその他の重要な情報を識別できます。

StackSet の説明

### パラメータ

パラメータは、テンプレートで定義されます。また、パラメータを使用すると、スタックを作成または更新する際にカスタム値を入力できます。

SampleParameter

# スタックセットの作成 2/2

## 3. スタックセットオプションの設定 タグや実行設定を指定。

### タグ

スタックのリソースに適用するタグ (キーと値のペア) を指定できます。スタックごとに一意のタグを 50 個まで追加できます。

スタックに関連付けられたタグがありません。

**新しいタグの追加**

さらに 50 のタグを追加できます

### 実行設定

#### マネージド型の実行

StackSets が競合しないオペレーションを並行して実行し、競合するオペレーションはキューに入れるかどうかを指定します。

非アクティブ  
StackSets は、一度に 1 つのオペレーションを実行します。

アクティブ  
StackSets は、競合しないオペレーションを並行して実行し、競合するオペレーションをキューに入れます。競合するオペレーションが終了すると、StackSets はリクエスト順にキューに入れられたオペレーションを開始します。

## 4. デプロイオプションの設定 デプロイ先や自動デプロイオプション、 同時実行するアカウントやリージョン等を指定。

### デプロイターゲット

StackSets は、ターゲット組織または組織単位 (OU) のすべてのアカウントにスタックインスタンスをデプロイします。親 OU をターゲットとして追加すると、StackSets はターゲットとして子 OU も追加します。 [詳細はこちら](#)

組織へのデプロイ  組織単位 (OU) へのデプロイ

### 自動デプロイオプション

**自動デプロイ**  
自動デプロイが有効になっている場合、アカウントが OU に追加されると、StackSets は自動的に追加のスタックインスタンスをこのアカウントにデプロイします。アカウントが OU から削除されると、StackSets はこのアカウントのスタックインスタンスを自動的に削除します。

有効  無効

**アカウント削除の動作**  
ターゲット OU からアカウントを削除する場合、アカウント内のスタックインスタンスを削除または保持する必要がありますか?

スタックを削除  スタックを保持

### リージョンの指定

スタックをデプロイするリージョンを選択します。スタックは、指定した順序でこれらのリージョンにデプロイされます。スタックセットの操作中、管理者アカウントとターゲットアカウントは、アカウント自体、ならびに関連するスタックセットおよびスタックセットインスタンスに関するメタデータを交換することに注意してください。 [詳細はこちら](#)

### デプロイオプション

**同時アカウントの最大数 - オプション**  
スタックを同時にデプロイできるリージョン別のアカウント数。数値が大きければ、オペレーションが高速になります。

数値

**障害耐性 - オプション**  
スタックが失敗できるリージョン別のアカウント数。この値を揃えると、このリージョンでのオペレーションが CloudFormation で停止されます。1 つのリージョンで停止されたオペレーションは、その他のリージョンでも続行されなくなります。数値が小さいほど、オペレーションの安全性が高くなります。

数値

**リージョンの同時実行**  
選択して、StackSets をリージョンに順次デプロイするか、並行してデプロイします。

順次  
リージョンのデプロイ順序を指定し、複数の StackSets オペレーションを同時に 1 つのリージョンにデプロイします。

並行  
指定したすべてのリージョンに複数の StackSets オペレーションを並行してデプロイします。

**同時実行モード**  
オペレーション実行中の同時実行レベルの動作を指定します。

厳格な耐障害性  
同時実行レベルを動的に下げて、失敗したアカウントの数が耐障害性 +1 を超えないようにします。

ソフト耐障害性  
常に指定された同時実行レベルで実行します。

# 自動デプロイオプション

組織単位（OU）へアカウントを追加・削除した際のスタックインスタンスの挙動を設定する機能。

## 自動デプロイ

自動デプロイが有効になっている場合、アカウントが OU に追加されると、StackSets は自動的に追加のスタックインスタンスをこのアカウントにデプロイします。アカウントが OU から削除されると、StackSets はこのアカウントのスタックインスタンスを自動的に削除します。

- 有効
- 無効

## アカウント削除の動作

ターゲット OU からアカウントを削除する場合、アカウント内のスタックインスタンスを削除または保持する必要がありますか？

- スタックを削除
- スタックを保持

※サービスマネージャのアクセス許可モデルの場合に設定することが可能。

※上書きされたパラメータ値[p.55、56]は、設定時に指定されたアカウントにのみ適用され、自動デプロイオプションにて今後追加されるアカウントには**適用されない**。

※自動デプロイ設定は、スタックセット作成後いつでも調整可能。

StackSet 名	StackSet ID
<a href="#">test-StackSet</a>	test-StackSet:69069fdd-e97d-4e52-8db5-8665e039ac79



自動デプロイ

自動デプロイが有効になっている場合、アカウントが OU に追加されると、StackSets は自動的に追加のスタックインスタンスをこのアカウントにデプロイします。アカウントが OU から削除されると、StackSets はこのアカウントのスタックインスタンスを自動的に削除します。

- 有効
- 無効

アカウント削除の動作

ターゲット OU からアカウントを削除する場合、アカウント内のスタックインスタンスを削除または保持する必要がありますか？

- スタックを削除
- スタックを保持

キャンセル 保存する

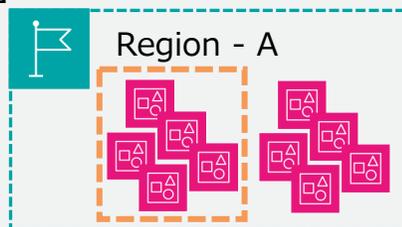
# デプロイオプション

## 同時アカウントの最大数

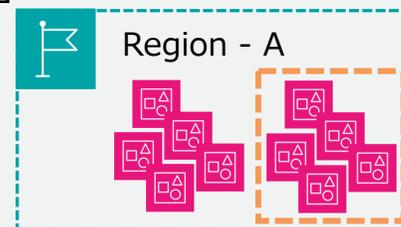
スタックセットの作成、更新、削除に適用され、リージョン別に一度にオペレーションを実行するターゲットアカウントの最大数または割合 (%) を指定することが可能。

例：割合 (%) : 50 で 2 つのリージョンの 10 個のターゲットアカウントにデプロイする場合

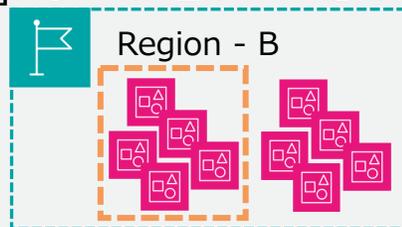
[1]. リージョン A で 5 アカウント



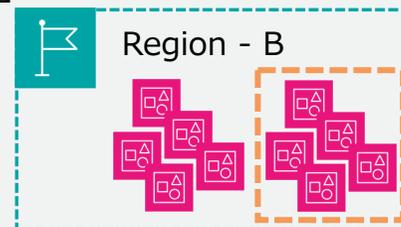
[2]. リージョン A で他の 5 アカウント



[3]. リージョン B で 5 アカウント



[4]. リージョン B で他の 5 アカウント



※指定された割合 (%) が指定したアカウントの整数にならない場合は、丸められる。

例：ターゲットアカウント 10 割合 (%) : 25 = ターゲットアカウントの最大数 2

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/stacksets-concepts.html#stackset-ops-options](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/stacksets-concepts.html#stackset-ops-options)

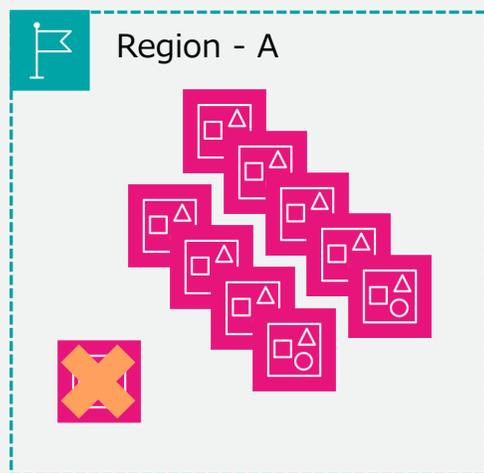
# デプロイオプション

## 障害耐性

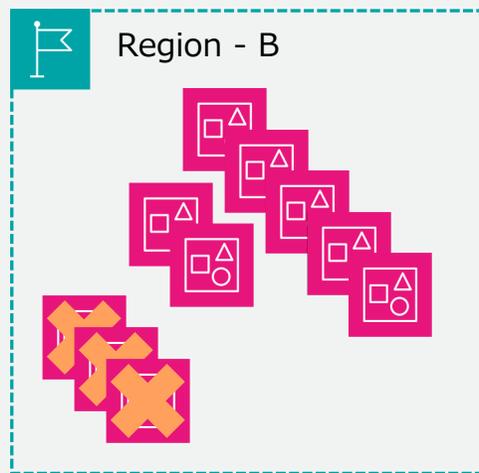
スタックセットの作成、更新、削除時に、各リージョンで発生する可能性があるスタックオペレーションの失敗の最大数または割合 (%) を超えるとオペレーションが自動停止する。

例：割合 (%)：20 で 3 つのリージョンの 10 個のターゲットアカウントにデプロイ

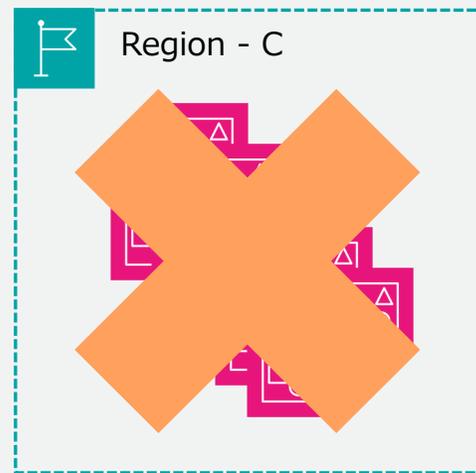
[1]. デプロイに 1 つ失敗



[2]. デプロイに 3 つ失敗



[3]. 更新オペレーション停止



※指定された割合 (%) が指定したアカウントの整数にならない場合は、丸められる。

例：ターゲットアカウント 10 割合 (%)：25 = ターゲットアカウントの最大数 2

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/stacksets-concepts.html#stackset-ops-options](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/stacksets-concepts.html#stackset-ops-options)

# デプロイオプション

## 同時実行モード

スタックセット操作中の同時実行レベルの動作を選択できるパラメータ。

### 厳密な障害耐性（デフォルト）

失敗したアカウントの数が**障害耐性 +1** の値を超えないように、同時実行レベルを動的に下げる。つまり、デプロイに失敗した場合、実際の同時実行数は失敗したデプロイの数に比例して減少する。そのため、デプロイが失敗するたびに同時実行性が低下するのでデプロイ速度は低下する。なお実際の同時実行数の初期値は、**同時アカウントの最大数**の値または**障害耐性 +1** の値のいずれか低い方に設定される。

### ソフト障害耐性

**障害耐性**と実際の同時実行性を切り離し、デプロイに失敗しても同時実行レベルは一定。これにより、障害の数に関係なくスタックセットの操作を**同時アカウントの最大数**の値で設定された同時実行レベルで実行できる。しかし、実際の障害の数を考慮せず同時にデプロイを実施するため、障害耐性で設定した数よりも、**デプロイに失敗したスタックインスタンスが多くなる可能性がある**ことに注意。そのため、厳密な障害耐性よりもデプロイ速度を優先したい場合に推奨。

# デプロイオプション

## 同時実行モード：厳密な障害耐性（デフォルト）

[例] 障害耐性：5 同時アカウントの最大数：10

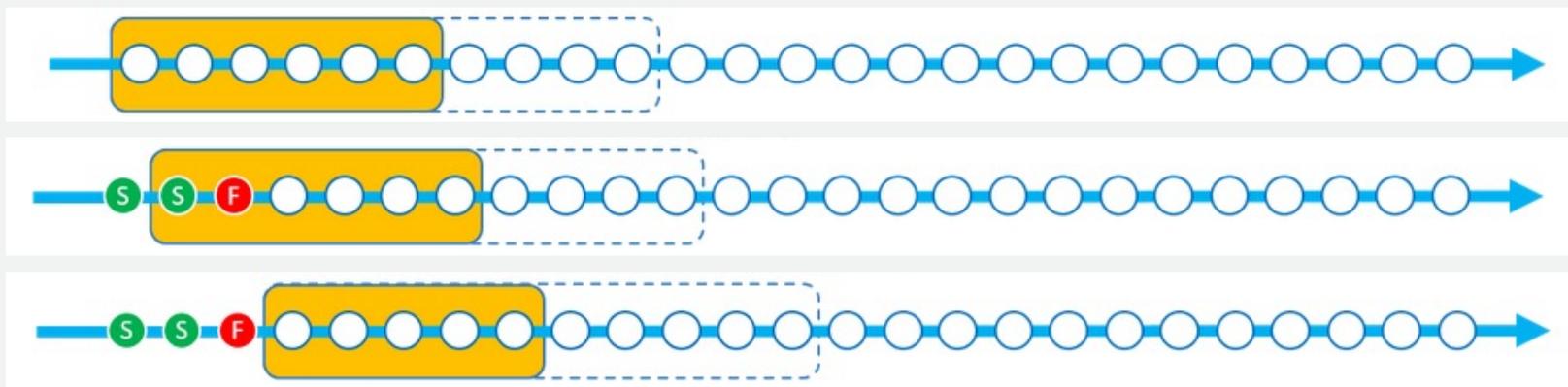
実際の同時実行数（障害耐性 + 1）：6

（障害耐性の値 5 + 1 が同時アカウントの最大数の値よりも低いため。）

 = 実際の同時実行数  = 同時アカウントの最大数

 = スタックセット  = 成功したスタックセット  = 失敗したスタックセット

(1) スタックセットが 1 つのデプロイに失敗すると、実際の同時実行数は 6 から 5 に減少する。



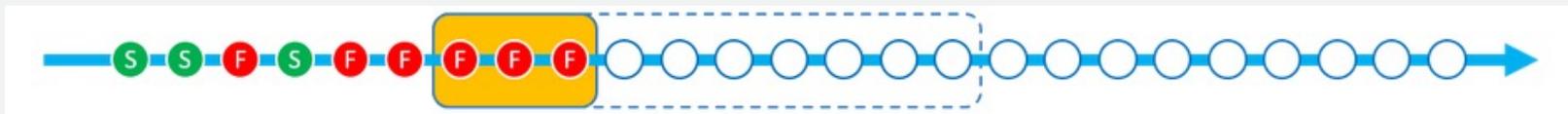
# デプロイオプション

## 同時実行モード：厳密な障害耐性（デフォルト）

- (2) スタックセットがさらに 2 つのスタックインスタンスのデプロイに失敗すると、実際の同時実行数は 5 から 3 に減少し、失敗したスタックインスタンスの合計は 3 となる。



- (3) その後、スタックセットが 3 つのスタックインスタンスのデプロイに失敗すると、失敗したスタックインスタンスの合計は 6 となる。  
その場合、失敗したスタックインスタンスの数が**障害耐性 +1** の値に等しくなるため、スタックセットは操作に失敗する。



今回の例では、スタックセットは操作を停止する前に 9 つのスタックインスタンス (3 つは成功、6 つは失敗) をデプロイを実施した。

# デプロイオプション

## 同時実行モード：ソフト障害耐性

[例] 障害耐性：5 同時アカウントの最大数：10  
実際の同時実行数：10

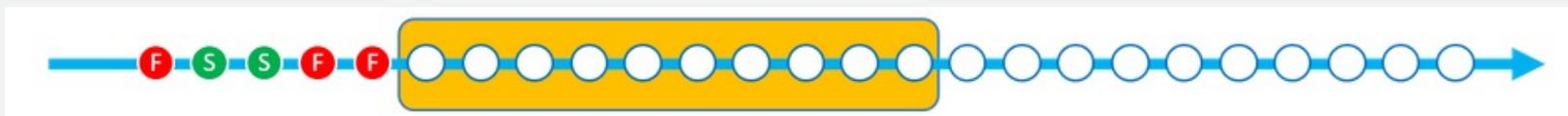
 = 実際の同時実行数  = 同時アカウントの最大数

○ = スタックセット    ●S = 成功したスタックセット    ●F = 失敗したスタックセット

(1) スタック操作が 1 つ失敗しても、実際の同時実行数は 10 のまま変わらない。



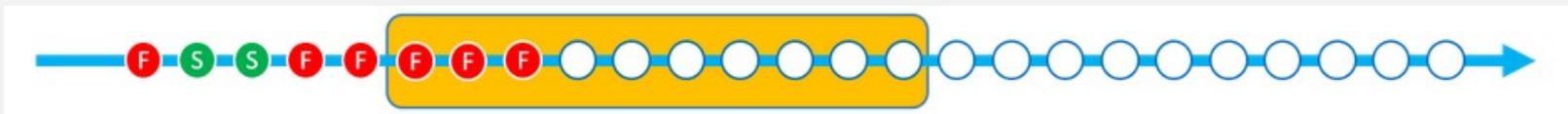
(2) さらに、スタックインスタンスに 2 回障害が発生しても、実際の同時実行数は 10 のまま変わらない。



# デプロイオプション

## 同時実行モード：ソフト障害耐性

- (3) 6 つのスタック操作に失敗し、**障害耐性**の値である 5 に達したため、スタックセットは操作に失敗する。ただし、**同時実行キューの残りの操作が終了するまでスタックセットの操作は終了しない。**



- (4) スタックセットの操作が**障害耐性**の 5 に達したにもかかわらず、同時実行キューに実行すべき操作が 7 つ残っていたため、合計 8 つのスタックインスタンスが失敗する。



この例では、スタックセットは操作を停止する前に 15 個のスタックインスタンス (7 つは成功、8 つは失敗) をデプロイした。

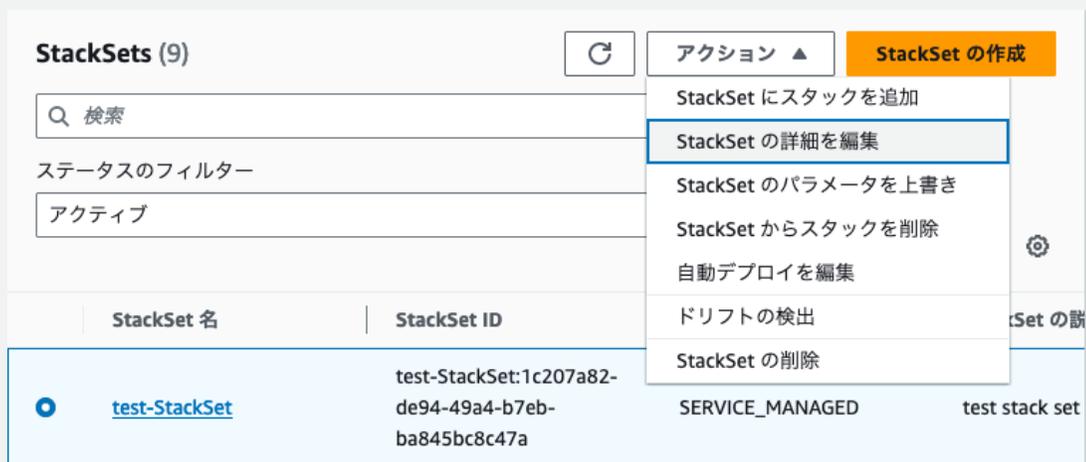
# オペレーション

- **アクセス許可モデルの選択と設定 (初回のみ)**
  - スタックセットはセルフマネージド型またはサービスマネージド型のアクセス許可が必要。
- **スタックセットの作成**
  - スタックの作成に使用するテンプレート、ターゲットアカウント、デプロイする AWS リージョンの指定を行い作成。
- **スタックセットの更新**
  - スタックセットを更新すると、スタックセットのスタックに変更内容がプッシュされる。
- **スタックインスタンスのパラメータの上書き**
  - アカウントおよびリージョン別にスタックインスタンスのパラメータ値を上書き可能。
- **スタックの削除**
  - 指定したリージョン内の指定したターゲットアカウントから、スタックを削除する。  
(スタックの削除に併せて対象のスタックインスタンスも削除される。)
- **スタックセットの削除**
  - スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。

# スタックセットの更新 1/2

## 1. 更新するスタックセットを選択

任意のスタックセットを選択し、アクションから「StackSet の詳細を編集」を押下。



The screenshot shows the AWS CloudFormation console interface. At the top, there's a search bar and a filter for 'アクティブ' (Active). Below that is a table of StackSets. The first entry is selected, and an action menu is open, highlighting 'StackSet の詳細を編集' (Edit StackSet details).

StackSet 名	StackSet ID	タイプ	ステータス
<a href="#">test-StackSet</a>	test-StackSet:1c207a82-de94-49a4-b7eb-ba845bc8c47a	SERVICE_MANAGED	test stack set

## 2. テンプレートの選択

現在のテンプレートを使用するか、既存のテンプレートを置き換えるか選択。

## 3. スタックセットの詳細を変更可能

スタックセットの説明、テンプレートのパラメータを変更可能。



The screenshot shows the 'StackSet の詳細を編集' (Edit StackSet details) page. It has several sections: 'StackSet 名' (StackSet name) with a text input field containing 'test-StackSet'; 'StackSet の説明' (StackSet description) with a text area containing 'test stack set'; and 'パラメータ' (Parameters) with a text area. There are also instructions for each field.

# スタックセットの更新 2/2

## 3. スタックセットオプションの設定

タグや実行設定を指定。

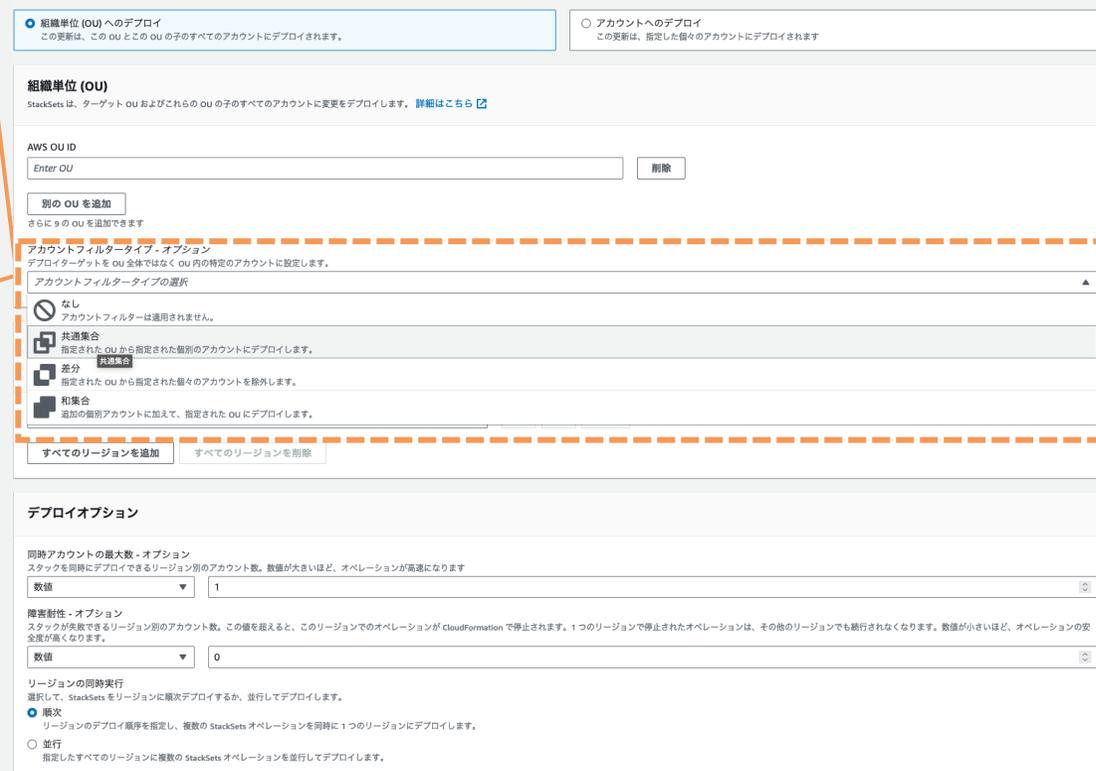
### アカウントフィルタータイプ(サービスマネージド)

デプロイターゲットを個々のアカウントに制限したり、指定された組織単位(OU)を使用してターゲットアカウントを指定することが可能。

	なし アカウントフィルターは適用されません。
	共通集合 指定された OU から指定された個別のアカウントにデプロイします。
	差分 指定された OU から指定された個々のアカウントを除外します。
	和集合 追加の個別アカウントに加えて、指定された OU にデプロイします。

## 4. デプロイオプションの設定

デプロイ先やリージョン、デプロイオプションやリージョン等を指定。



The screenshot shows the 'Deploy to' section of the AWS CloudFormation console. It includes two radio buttons at the top: 'Deploy to organization (OU)' (selected) and 'Deploy to account'. Below this is the 'Organization (OU)' section, which contains an 'AWS OU ID' input field with a 'Delete' button and a 'Add other OUs' button. A dropdown menu for 'Account filter type' is highlighted with a dashed orange box, showing options: 'None', 'Intersection', 'Difference', and 'Union'. Below this are buttons for 'Add all regions' and 'Remove all regions'. The 'Deployment options' section includes a 'Maximum number of accounts to deploy simultaneously' dropdown set to '1', a 'Fault tolerance' dropdown set to '0', and radio buttons for 'Sequential' (selected) and 'Parallel' deployment across regions.

[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/account-level-targets.html](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/account-level-targets.html)  
[https://docs.aws.amazon.com/ja\\_jp/AWSCloudFormation/latest/UserGuide/stacksets-update.html](https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/stacksets-update.html)

# オペレーション

- **アクセス許可モデルの選択と設定 (初回のみ)**
  - スタックセットはセルフマネージド型またはサービスマネージド型のアクセス許可が必要。
- **スタックセットの作成**
  - スタックの作成に使用するテンプレート、ターゲットアカウント、デプロイする AWS リージョンの指定を行い作成。
- **スタックセットの更新**
  - スタックセットを更新すると、スタックセットのスタックに変更内容がプッシュされる。
- **スタックインスタンスのパラメータの上書き**
  - アカウントおよびリージョン別にスタックインスタンスのパラメータ値を上書き可能。
- **スタックの削除**
  - 指定したリージョン内の指定したターゲットアカウントから、スタックを削除する。  
(スタックの削除に併せて対象のスタックインスタンスも削除される。)
- **スタックセットの削除**
  - スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。

# スタックインスタンスのパラメータの上書き 1/2

## 1. 更新するスタックセットを選択

任意のスタックセットを選択し、アクションから「StackSet のパラメータを上書き」を押下。

StackSet 名	StackSet ID	ステータス
<a href="#">test-StackSet</a>	test-StackSet:69069fdd-e97d-4e52-8db5-8665e039ac79	SERVICE_MANAGED

## 2. デプロイオプションの設定

デプロイ先やリージョン、デプロイオプションやリージョン等を指定。

組織単位 (OU) へのデプロイ  
この更新は、この OU とこの OU の子すべてのアカウントにデプロイされます。

アカウントへのデプロイ  
この更新は、指定した個々のアカウントにデプロイされます。

**組織単位 (OU)**  
StackSets は、ターゲット OU およびこれらの OU の子すべてのアカウントに変更をデプロイします。 [詳細はこちら](#)

AWS OU ID  
Enter OU

さらに 9 の OU を追加できます

アカウントフィルタータイプ - オプション  
デプロイターゲットを OU 全体ではなく OU 内の特定のアカウントに設定します。  
アカウントフィルタータイプの選択

**リージョンの指定**  
スタックをデプロイするリージョンを選択します。スタックは、指定した順序でこれらのリージョンにデプロイされます。スタックセットの操作中に、管理アカウントとターゲットアカウントは、アカウント自体、ならびに関連するスタックセットおよびスタックセットインスタンスに関するメタデータを交換することに注意してください。 [詳細はこちら](#)

**デプロイオプション**

同時アカウントの最大数 - オプション  
スタックを同時にデプロイできるリージョン別のアカウント数。数値が大きほど、オペレーションが高速になります

数値

障害耐性 - オプション  
スタックが失敗できるリージョン別のアカウント数。この値を越えると、このリージョンでのオペレーションが CloudFormation で停止されます。1つのリージョンで停止されたオペレーションは、その他のリージョンでも実行されなくなります。数値が小さいほど、オペレーションの安全度が高くなります。

数値

リージョンの同時実行  
選択して、StackSets をリージョンに順次デプロイするか、並行してデプロイします。

順次  
リージョンのデプロイ順序を指定し、複数の StackSets オペレーションを同時に 1 つのリージョンにデプロイします。

並行  
指定したすべてのリージョンに複数の StackSets オペレーションを並行してデプロイします。

同時実行モード  
オペレーション実行中の同時実行レベルの動作を指定します。

厳格な耐障害性  
同時実行レベルを動的に下げて、失敗したアカウントの数が耐障害性 + 1 を超えないようにします。

ソフト耐障害性  
常に指定された同時実行レベルで実行します。

# スタックインスタンスのパラメータの上書き 2/2

## 3. 上書きの指定

任意のパラメータを選択し、アクションから「StackSet 値の上書き」を押下。

名前	StackSet 値	アクション
<input checked="" type="checkbox"/> MultiRegion	false	<b>StackSet 値の上書き</b> StackSet 値に設定 上書きフィールドを元に戻す

上書きする内容を入力することが可能。

StackSet のパラメータ値を上書き

Trail Configuration

Is this a multi-region trail  
Indicates whether the CloudTrail trail is created in the region in which you create the stack (false) or in all regions (true).

String を選択

Q |

true

false

キャンセル 変更の保存

## 留意点

※上書きされたパラメータをスタックセットで指定された値に戻すためには、「StackSet 値に設定」を選択する必要がある。

名前	StackSet 値	値の上書き
<input checked="" type="checkbox"/> MultiRegion	false	<b>StackSet 値に戻す</b>

※上書きされたパラメータ値は、設定時に指定したアカウントにのみ適用される。  
(今後作成されるスタックには**適用されない**。)

# オペレーション

- **アクセス許可モデルの選択と設定 (初回のみ)**
  - スタックセットはセルフマネージド型またはサービスマネージド型のアクセス許可が必要。
- **スタックセットの作成**
  - スタックの作成に使用するテンプレート、ターゲットアカウント、デプロイする AWS リージョンの指定を行い作成。
- **スタックセットの更新**
  - スタックセットを更新すると、スタックセットのスタックに変更内容がプッシュされる。
- **スタックインスタンスのパラメータの上書き**
  - アカウントおよびリージョン別にスタックインスタンスのパラメータ値を上書き可能。
- **スタックの削除**
  - 指定したリージョン内の指定したターゲットアカウントから、スタックを削除する。  
(スタックの削除に併せて対象のスタックインスタンスも削除される。)
- **スタックセットの削除**
  - スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。

# スタックの削除

## 1. 削除するスタックを選択

スタックセットを選択し、アクションから「StackSet からスタックを削除」を押下。

The screenshot shows the AWS CloudFormation console interface. At the top, there's a search bar and a filter dropdown set to 'アクティブ'. Below that is a table of StackSets. One StackSet is selected, and an 'アクション' (Actions) menu is open, showing 'StackSet からスタックを削除' (Delete stacks from StackSet) as the selected option.

StackSet 名	StackSet ID
<a href="#">test-StackSet</a>	test-StackSet:1c207a82-de94-49a4-b7eb-ba845bc8c47a

## 2. デプロイオプションの設定

デプロイ先やリージョン、デプロイオプションやリージョン等を指定。

The screenshot shows the 'デプロイオプション' (Deployment Options) configuration screen. It includes sections for '組織単位 (OU)' (Organization Unit), 'リージョンの指定' (Region Selection), and 'デプロイオプション' (Deployment Options). The 'デプロイオプション' section has several settings: '同時アカウントの最大数 - オプション' (Maximum number of concurrent accounts - Option) set to 1, '障害耐性 - オプション' (Fault tolerance - Option) set to 0, 'スタックを保持' (Keep stacks) checked, 'リージョンの同時実行' (Concurrent execution by region) set to '順次' (Sequential), and '同時実行モード' (Concurrent execution mode) set to '最適な障害耐性' (Optimal fault tolerance).

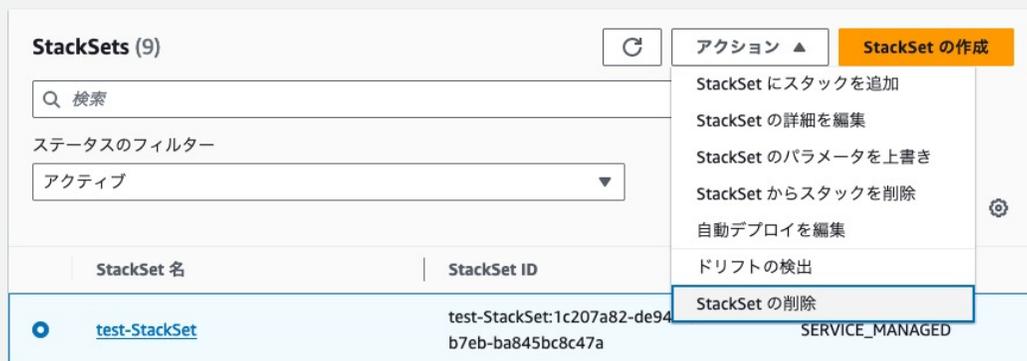
# オペレーション

- **アクセス許可モデルの選択と設定 (初回のみ)**
  - スタックセットはセルフマネージド型またはサービスマネージド型のアクセス許可が必要。
- **スタックセットの作成**
  - スタックの作成に使用するテンプレート、ターゲットアカウント、デプロイする AWS リージョンの指定を行い作成。
- **スタックセットの更新**
  - スタックセットを更新すると、スタックセットのスタックに変更内容がプッシュされる。
- **スタックインスタンスのパラメータの上書き**
  - アカウントおよびリージョン別にスタックインスタンスのパラメータ値を上書き可能。
- **スタックの削除**
  - 指定したリージョン内の指定したターゲットアカウントから、スタックを削除する。  
(スタックの削除に併せて対象のスタックインスタンスも削除される。)
- **スタックセットの削除**
  - スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。

# スタックセットの削除

## 1. 削除するスタックセットを選択

スタックセットを選択し、アクションから「StackSet の削除」を押下。



The screenshot shows the AWS CloudFormation console interface. At the top, there's a search bar and a filter dropdown set to 'アクティブ'. Below that is a table with columns 'StackSet 名' and 'StackSet ID'. One entry is selected: 'test-StackSet' with ID 'test-StackSet:1c207a82-de94b7eb-ba845bc8c47a'. An 'アクション' (Actions) menu is open over the selected row, listing various actions. The 'StackSet の削除' (Delete StackSet) option is highlighted.

StackSet 名	StackSet ID
<a href="#">test-StackSet</a>	test-StackSet:1c207a82-de94b7eb-ba845bc8c47a

## 2. 削除を実行

削除ボタンを押下。



The screenshot shows a confirmation dialog box titled 'StackSet を削除しますか?' (Delete StackSet?). The main text asks: 'スタックセット test-StackSet を完全に削除しますか? このアクションは元に戻すことができません' (Do you want to completely delete the StackSet test-StackSet? This action cannot be undone). At the bottom right, there are two buttons: 'キャンセル' (Cancel) and '削除' (Delete).

※スタックセット内にスタックインスタンスが存在しない場合のみ、削除可能。



Thank you!

# AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
  - <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBBlqY>



ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください  
#awsblackbelt

# 内容についての注意点

- 本資料では資料作成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。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://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
- 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)