

T4-1

クラウド移行への第一歩 AWSへの移行プロジェクトを始めてみよう

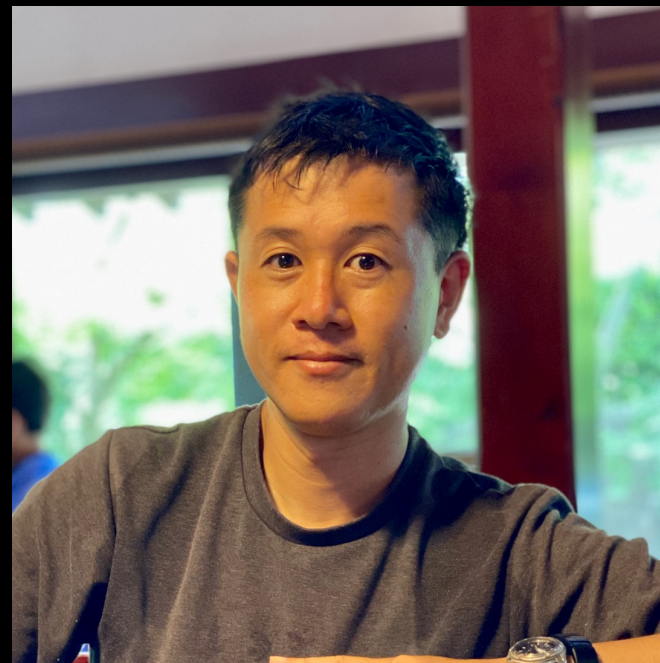
五十嵐 建平

プリンシパルソリューションアーキテクト

アマゾン ウェブ サービス ジャパン合同会社
技術統括本部

自己紹介

五十嵐 建平
(いがらし けんぺい)



2017年～ AWS にてソリューションアーキテクト
コンシューマ領域のお客様を担当
日本における Amazon Neptune の担当もやっています



クラウドへの移行の悩み

何のために移行するのか？



移行をどう計画すればいいのか？



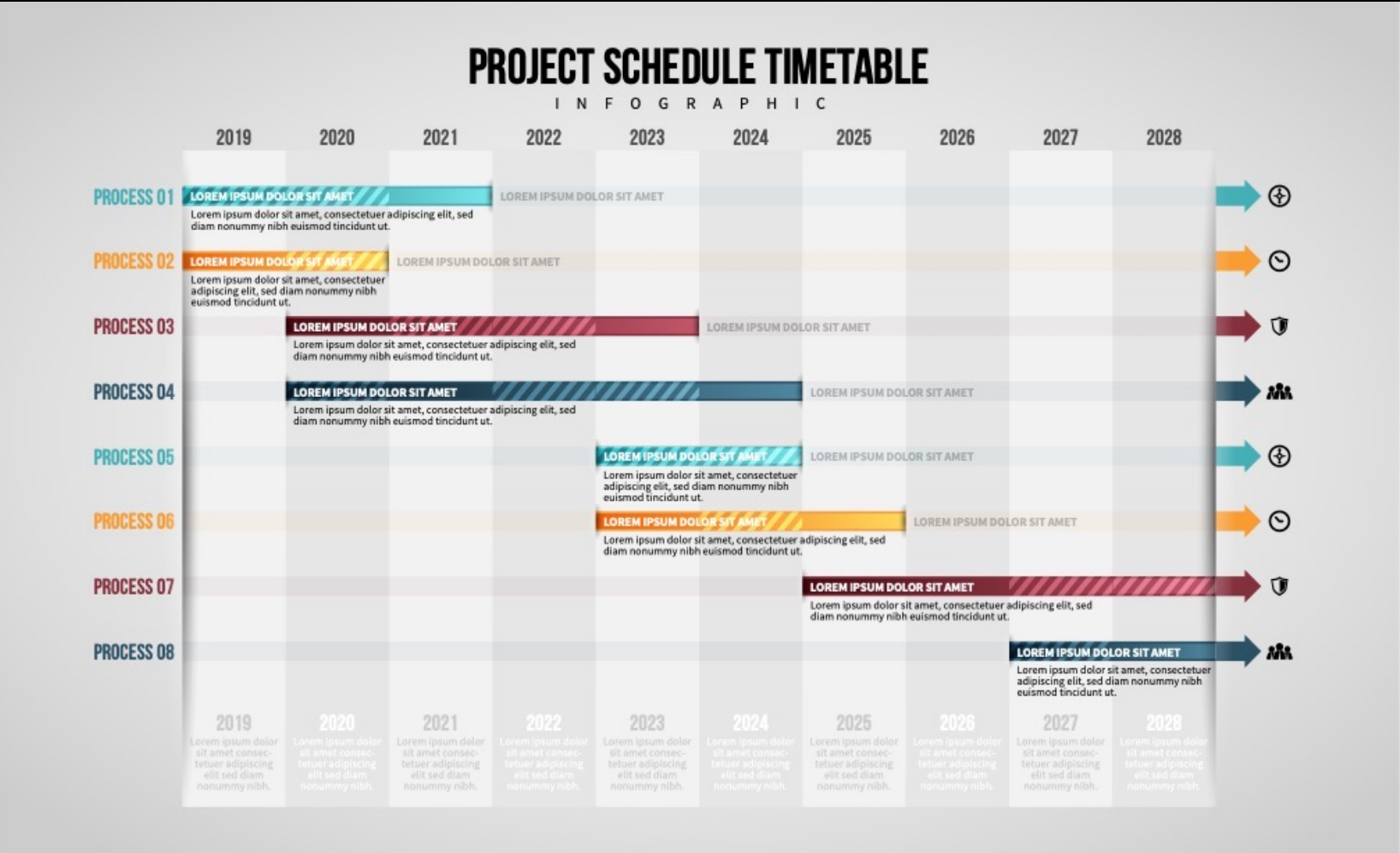
どんな構成にすればいいのか？



移行に便利なツールはどれ？



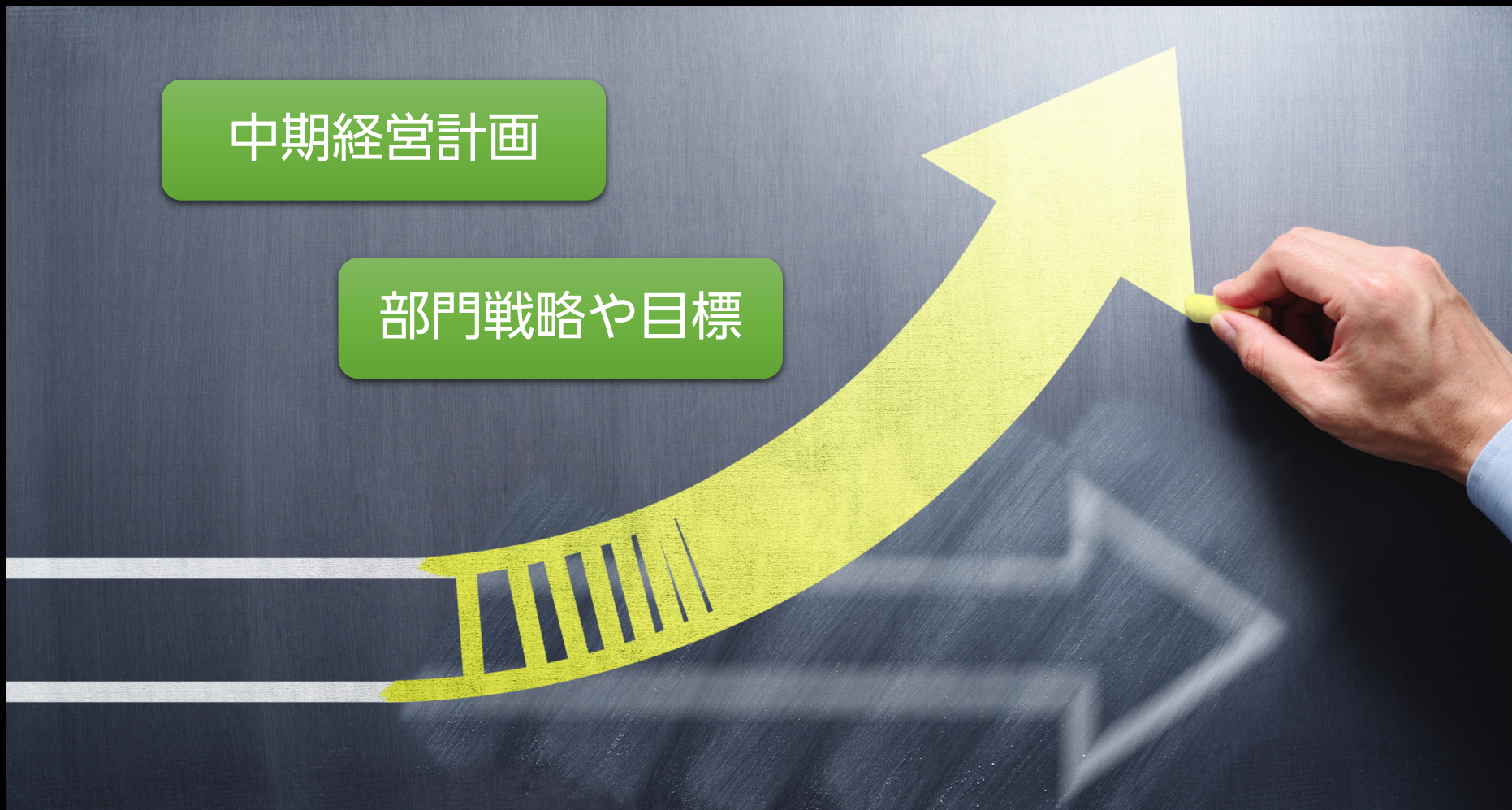
移行プロジェクトをどう進めるのか？



何のために移行するのか？



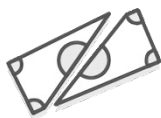
自社のビジネスに沿った目的の設定



移行目的の6つの柱と突合



コストダウン



耐障害性



俊敏性



運用負荷低減



グローバル展開



イノベーション



具体的なKPIを設定



コストダウン

x年でコストを
x%減らす

耐障害性

大阪への
DR環境を作る

俊敏性

リリース頻度を
年xx回に増やす

運用負荷低減

運用時間を
xx時間短縮する

グローバル展開

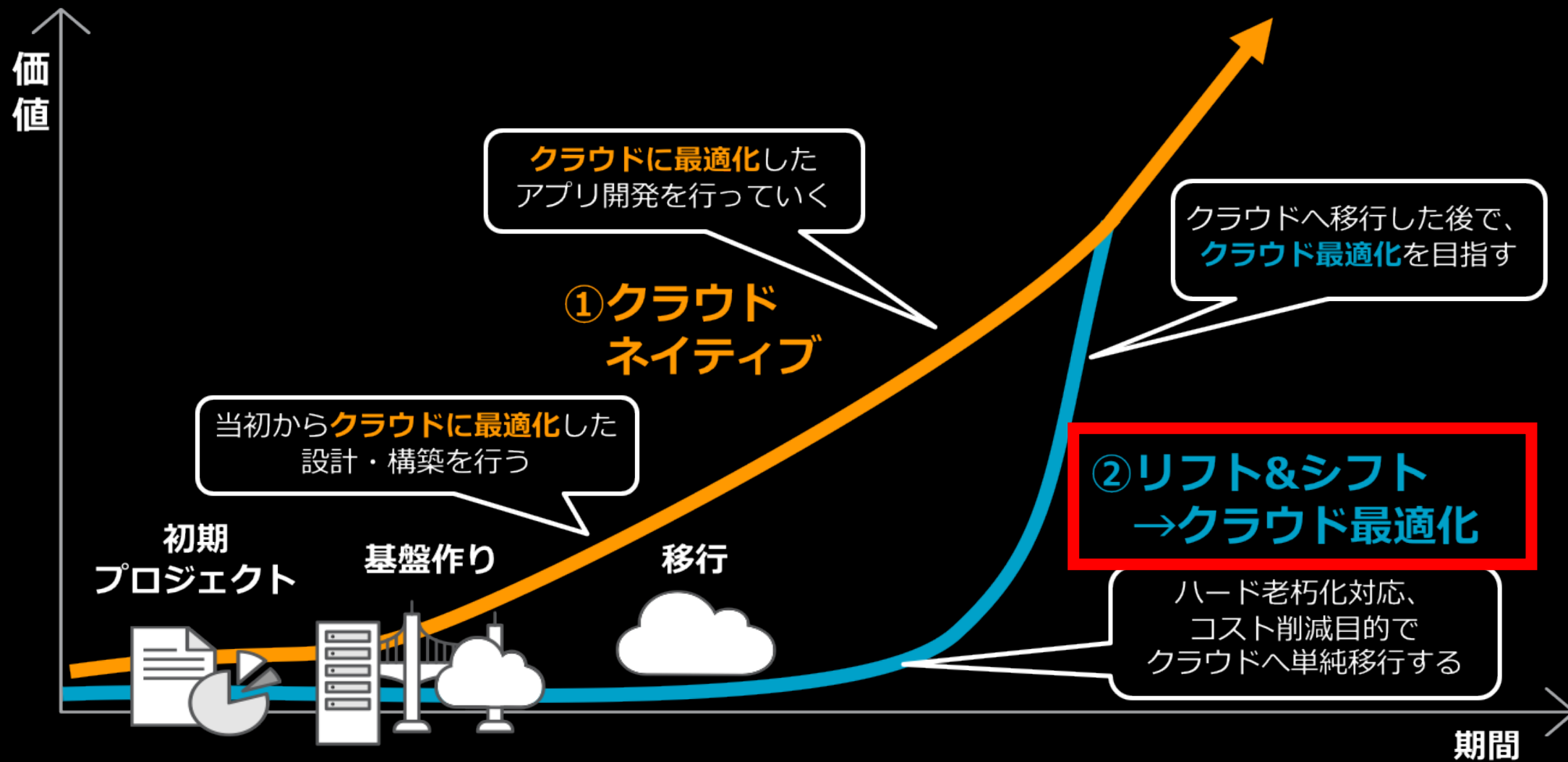
xx地域で
サービスを開始する

イノベーション

x個の新アプリを
ローンチさせる

移行をどう計画すればいいのか？

リフト&シフトを検討する



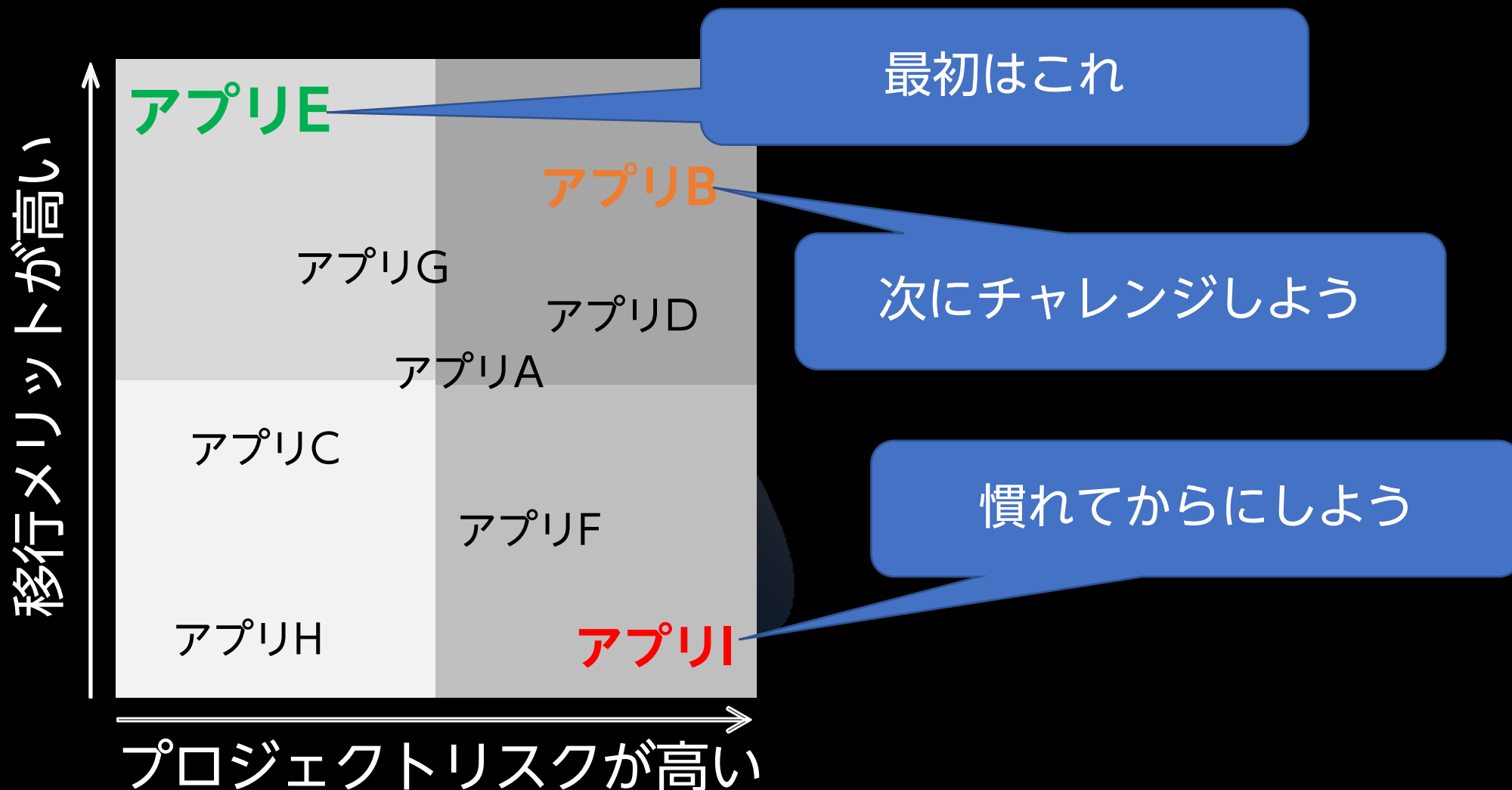
アプリケーションを分類してグループ化

業務別の分類

評価/分類	業務/アプリケーション・ドメイン								
	マーケティング		営業支援		物流		生産		...
	アプリ名	予算	アプリ名	予算	アプリ名	予算	アプリ名	予算	
戦略的	アプリI	¥xxx	アプリC アプリN	¥xxx ¥xxx	アプリE	¥xxx	アプリA アプリD	¥xxx	
新規	アプリZ	¥xxx	アプリX アプリY	¥xxx ¥xxx			<h3>特性別の分類</h3> <ul style="list-style-type: none"> ピーク性あり ¥xxxキャパプラが困難 一時的な利用 停止の可能性あり 		
一時/季節	アプリM アプリO アプリQ アプリR	¥xxx ¥xxx ¥xxx ¥xxx	アプリH アプリJ	¥xxx ¥xxx					
レガシー			アプリP	¥xxx	アプリF アプリL	¥xxx ¥xxx	アプリB アプリG アプリK	¥xxx ¥xxx ¥xxx	

アプリケーションの移行する順序を決める

影響度評価の例



どんな構成にすればいいのか？

アプリケーションの移行方式を決める

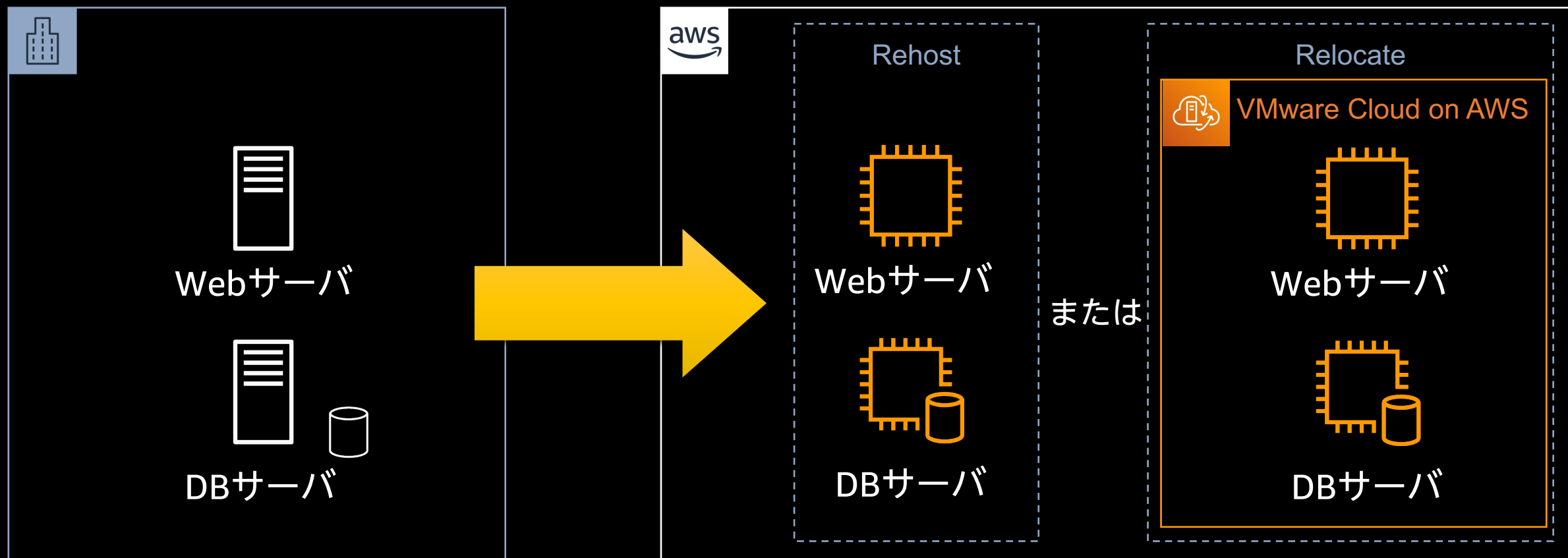
リフト &
シフト

クラウド
ネイティブ

移行方式	概要
Rehost	OSやアプリケーションをそのまま移行
Relocate	VMware環境をそのまま移行
Replatform	OSやDBを変更/アップグレードして移行
Refactor	クラウドネイティブに書き換え
Retire	サーバやアプリケーションを廃止する
Retain	オンプレミス環境で引き続き運用する
Repurchase	アプリケーションの買い替え

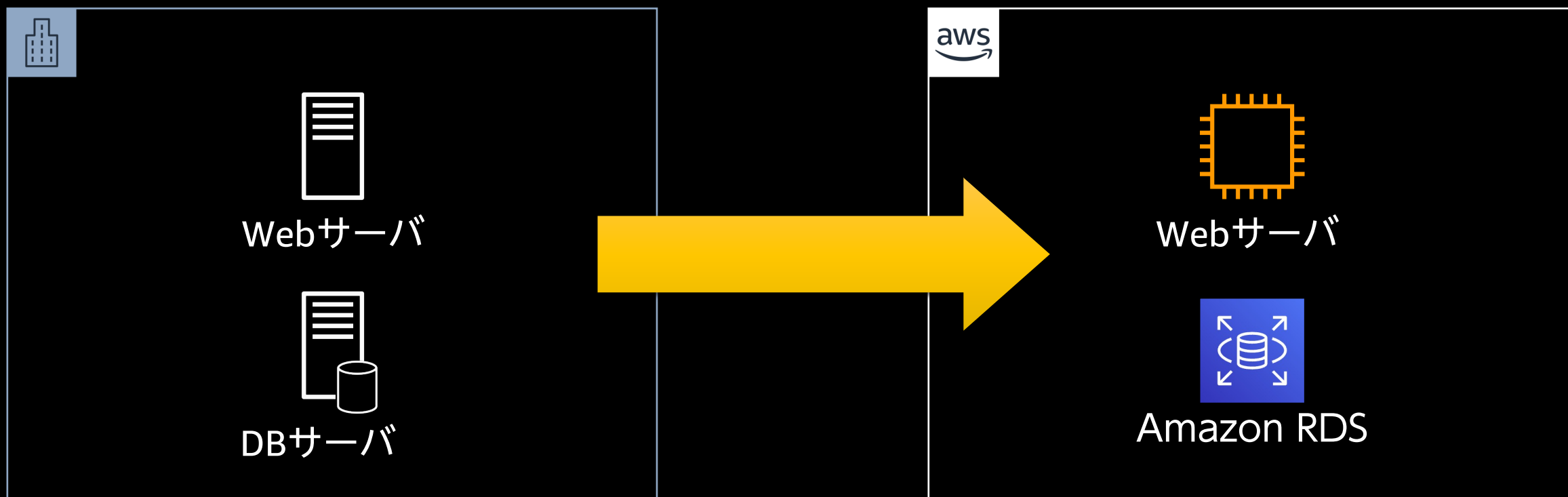
Rehost / Relocate: 単純移行

OSやアプリケーション、データベースに変更を加えずそのまま移行



Replatform: プラットフォームの更新

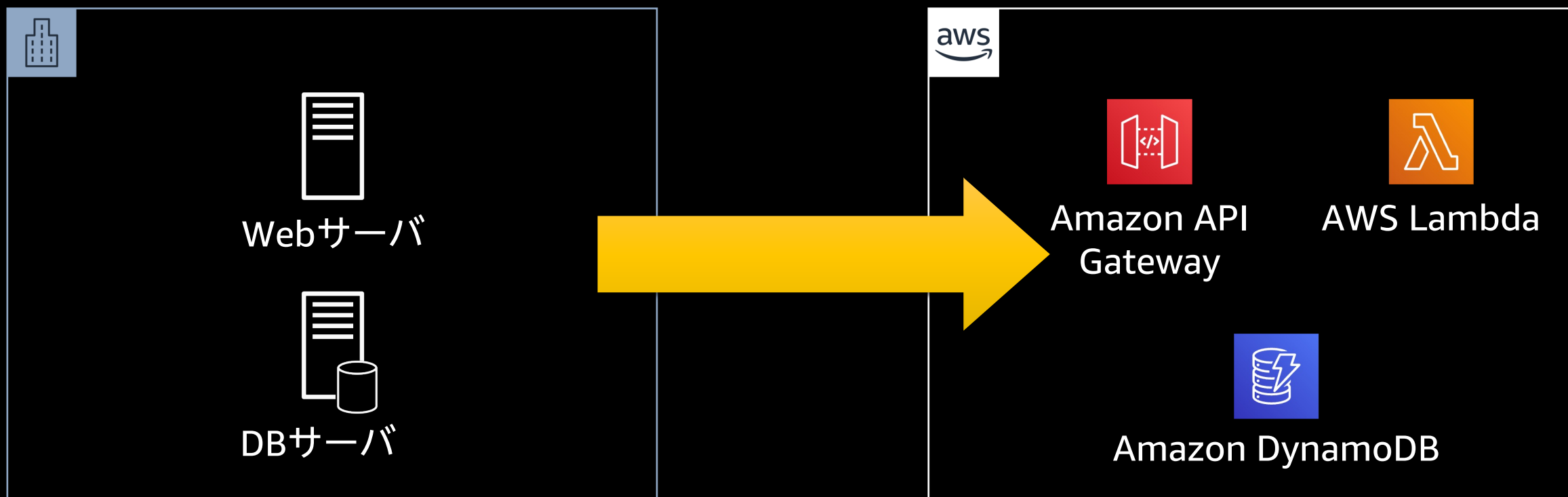
OSまたはDBの変更やアップグレード



Refactor: アプリケーションの書き換え

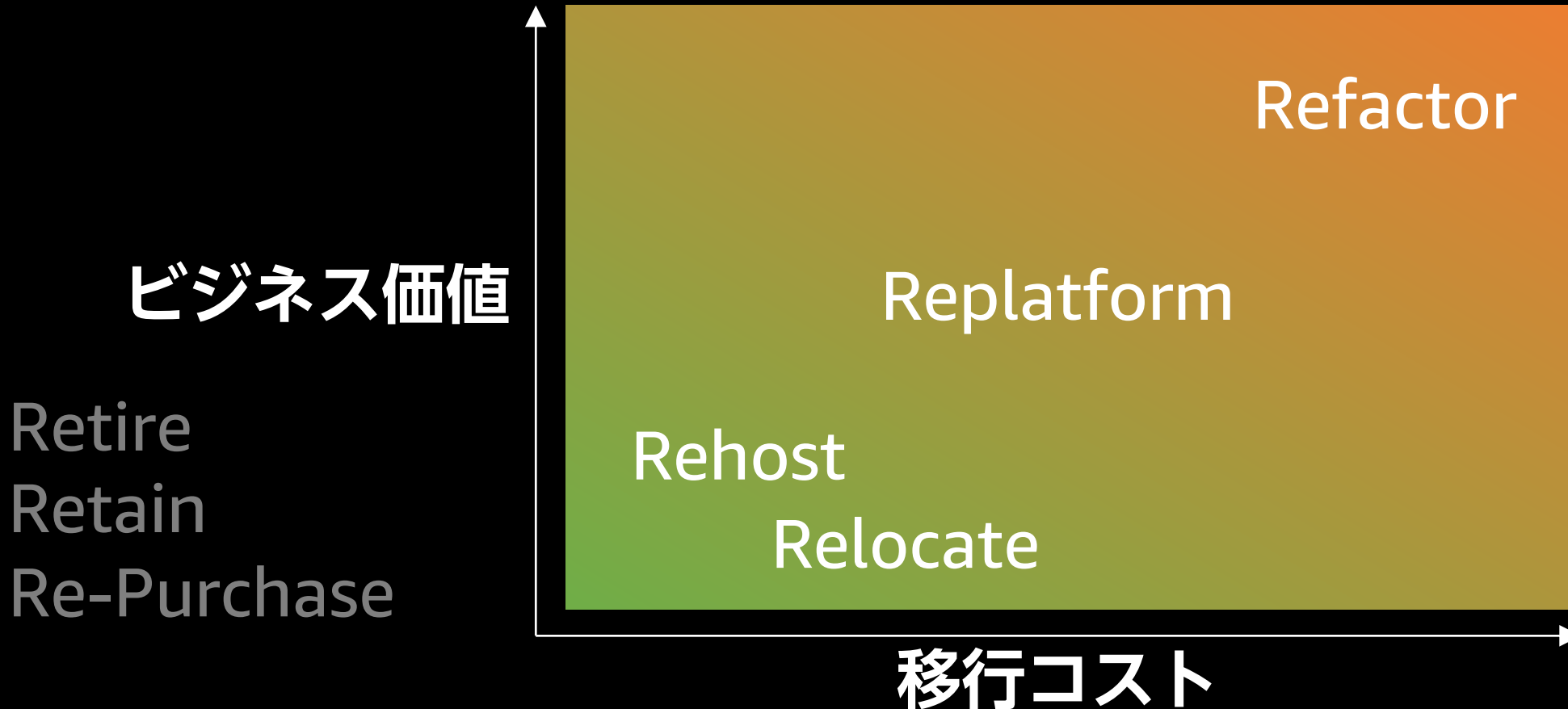
クラウドネイティブ

クラウドネイティブなアプリケーションへの書き換え



全体の移行方式を俯瞰する

ビジネス価値とコストから改めて全体の移行方式を決める



2つのフレームワーク

Cloud Adoption Framework

- ✓ 効果的なクラウドの導入を促進
- ✓ AWSの経験に基づく6つのパースペクティブ

Business People Governance
Platform Security Operations

<https://aws.amazon.com/jp/professional-services/CAF/>

Well-Architected Framework

- ✓ クラウドにおける設計の考え方におけるベストプラクティス
- ✓ 一貫した6つの評価軸を提供



<https://aws.amazon.com/jp/architecture/well-architected/>

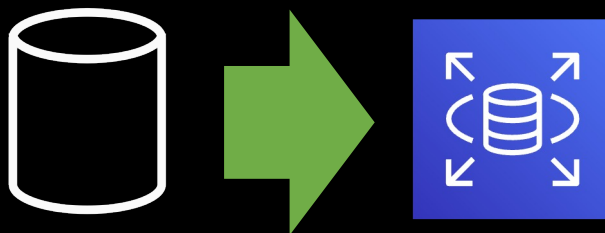
移行に便利なツールはどれ？

移行対象は3つ

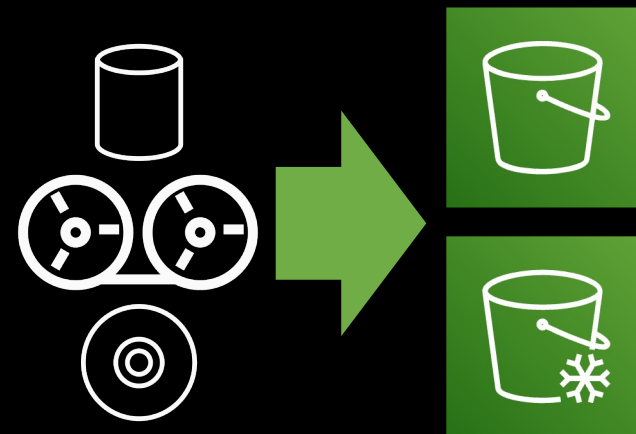
サーバ移行



データベース移行



データ移行



AWS Application Migration Service (MGN)



物理マシンおよび仮想マシンを移行するサービス

- サーバの増分変更を自動転送
- OSの種類やバージョンを変えずそのまま移行



AWS Database Migration Service (DMS)



データベースを最小限のダウンタイムで移行できるサービス

- データベースの増分変更を自動転送
- 異なるDBプラットフォームの移行にも対応

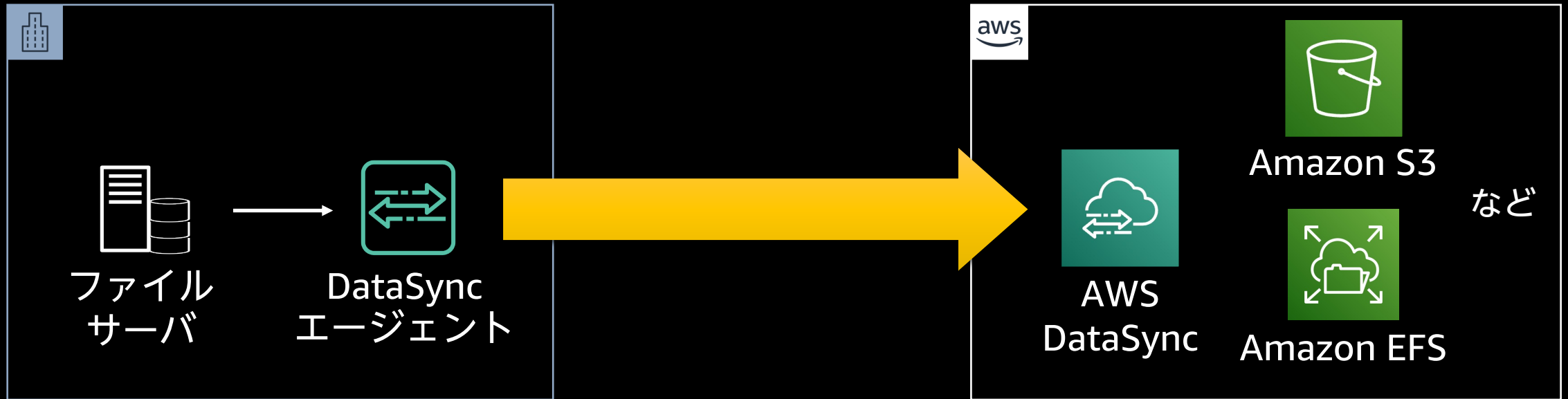


AWS DataSync



オンプレミスのファイルをクラウドへ転送するサービス

- オンプレミスのデータを高速、自動的に転送
- エージェントあたり最大10Gbpsでの転送が可能



AWS Snowball Edge / Snowcone



オンプレミスからクラウドへ大量のデータを移行するためのサービス

- 8TB～80TBの容量をもつ物理筐体を配送
- 回線に依存しないデータ転送を実現



移行ツールのまとめ

サーバ移行



AWS Application
Migration Service

データベース移行



AWS Database
Migration Service

データ移行



AWS DataSync



AWS Snowball Edge



AWS Snowcone

移行プロジェクトをどう進めるのか？

オンプレミスでは制約が多かった

オンプレミスでのプロジェクトの進め方

リソース拡張や
変更の制約

一時利用や
除却が困難

生産性の限界

働き方の制約

ソフト面の制約

システムリソース



マニュアルオペレーション



建物やスペース



ハード面の制約

オンプレミスでは机上検討がとても大事だった

調達してからの手戻りが発生しないように

要件定義

設計

実装
構築

テスト

運用



制約からの解放

AWS上に移行することで制約から解放される

俊敏性



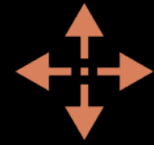
数分で確保できる
ITリソース

コスト削減



初期投資不要
従量課金

弾力性



サイジングからの解放

重要な作業に集中



本来の業務にフォーカス

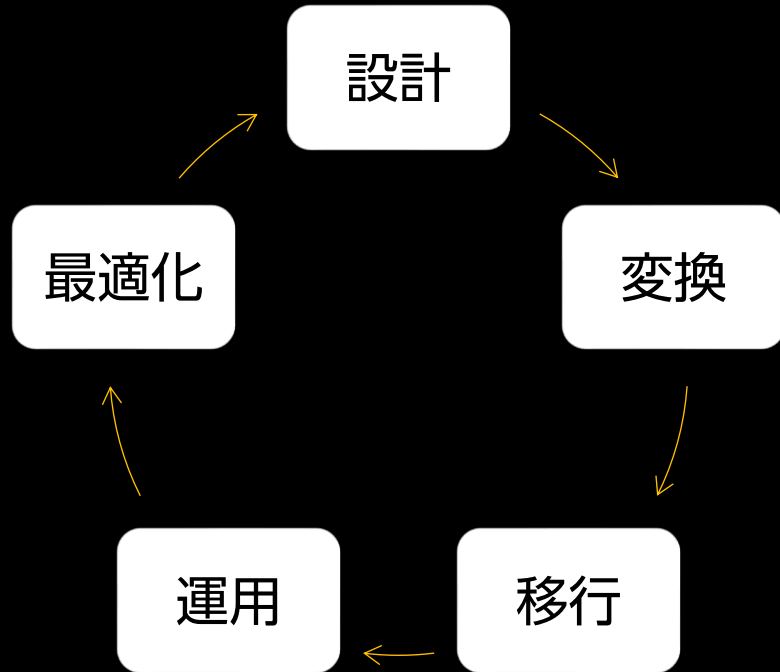
分単位で世界中にデプロイ



市場に出すまでの時間を短縮

実験を繰り返す

不明点は試してみても移行の要件と設計を固めていく



動くものを確認しながら進めるのが大切

机上でのみ検討するのではなく、実際に試す

- 実際に触ることでイメージが明確になる
- 早めに想定の違いが確かめられればすぐにやり直せる



推測せずに実測する

机上でのみ検討するのではなく、実際に試して測る

- 実測値に基づいた性能上限の把握
- 現実的な目標復旧時間(RTO)



必要な時にだけリソースを使う

従量課金を活用する

- 通常は機能テストが実施できる程度の最小限のリソース
- 休日は開発環境を停止
- 環境面の増減で作業を効率化



インフラの構成や設定をコード化

Infrastructure as Code(IaC) – インフラをコードとして扱う

- インフラをソースコードと同じように扱う
- 変更作業や複製作業の効率化
- ミスの発生確率の低下



AWS CloudFormation

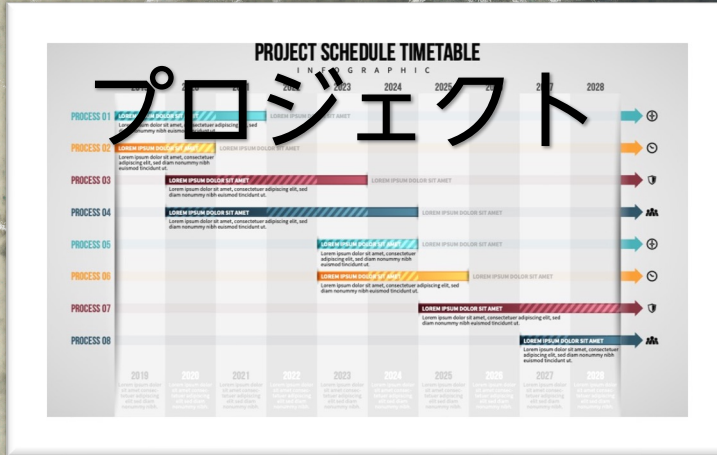
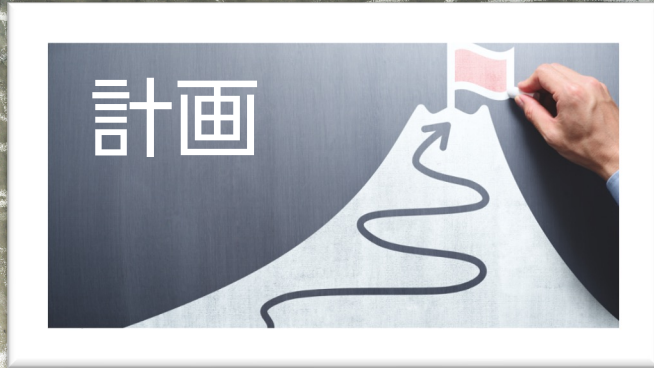


AWS Cloud Development Kit
(AWS CDK)

```
1 AWSTemplateFormatVersion: '2010-09-09'
2 Description: sample CFn template with YAML – Aurora
3
4 Parameters:
5   BaseStackName:
6     Description: VPC Stack Name
7     Type: String
8     MinLength: 1
9     MaxLength: 255
10    AllowedPattern: ^[a-zA-Z][~a-zA-Z0-9]*$
11    Default: yaml-stack2
12
13   SecStackName:
14     Description: Security Stack Name
15     Type: String
16     MinLength: 1
17     MaxLength: 255
18     AllowedPattern: ^[a-zA-Z][~a-zA-Z0-9]*$
19     Default: yaml-stack2-nw
20
21   DBUser:
22     Description: Database Master User Name
23     Type: String
24     MinLength: 5
25     MaxLength: 255
26     Default: dbuser
27
28   DBPassword:
29     Description: Database Master User Password
30     Type: String
31     MinLength: 5
32     MaxLength: 255
33     Default: dbpassword
34
35 Resources:
36   AuroraCluster:
37     Type: AWS::RDS::DBCluster
38     DependsOn: AuroraMonitorRole
39     DeletionPolicy: Snapshot
40     Properties:
41       DatabaseName: AuroraTest
42       Engine: aurora
43       MasterUsername: !Ref DBUser
44       MasterUserPassword: !Ref DBPassword
45       DBSubnetGroupName: !Ref AuroraSubnetGrp
46       VpcSecurityGroupIds:
47         - Fn::ImportValue: !Sub ${SecStackName}-SecGrpDB
48
49
```

まとめ





クラウドへの移行の悩みを
少しでも解決できていれば幸いです



AWS TRAINING & CERTIFICATION

AWS Skill Builder の 500+ の 無料デジタルコースで学ぼう

30以上のAWSソリューションの中から、自分に最も関係のあるクラウドスキルとサービスにフォーカスし、自習用のデジタル学習プランとRamp-Upガイドで学ぶことができます。

- 自分のペースでAWSクラウド上を活用した未来を切り開く
- 学習プランでスキルや知識を向上
- AWS認定資格でクラウドの専門知識を証明する

自分に合ったスキルアップ方法を学びましょう
[EXPLORE.SKILLBUILDER.AWS](https://explore.skillbuilder.aws) »



AWS Builders Online Series にご参加いただきありがとうございます

楽しんでいただけましたか? ぜひアンケートにご協力ください。
本日のイベントに関するご意見/ご感想や今後のイベントについてのご希望や改善のご提案などがございましたら、ぜひお聞かせください。



aws-apj-marketing@amazon.com



twitter.com/awscloud_jp



facebook.com/600986860012140



<https://www.youtube.com/user/AmazonWebServicesJP>



<https://www.linkedin.com/showcase/aws-careers/>



twitch.tv/aws

Thank you!

