



# AWS Migration Hub Refactor Spaces

Shingo Maeda

# モノリシックアプリケーションのリファクタリングは難易度が高い



## ビジネス

- 製品のアップデートが遅い
- マーケットへの展開に時間がかかる
- 各サブシステム担当チームの責任境界が曖昧
- 一部の変更がシステム全体及ぼす影響が予測困難



## 技術

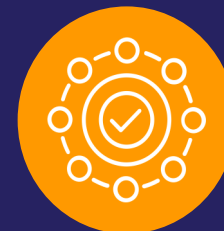
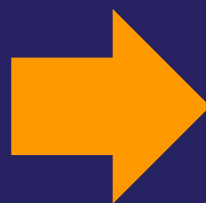
- 高い結合度と低い凝集度
- デプロイが難しく低頻度にならざるを得ない
- アプリケーションが不安定で理解困難
- ちょっとした試みが大きなインパクトになる可能性

# 多くのお客様が 2段階 を経てモダナイズ化



## フェーズ1 lift and shift(リホスト)

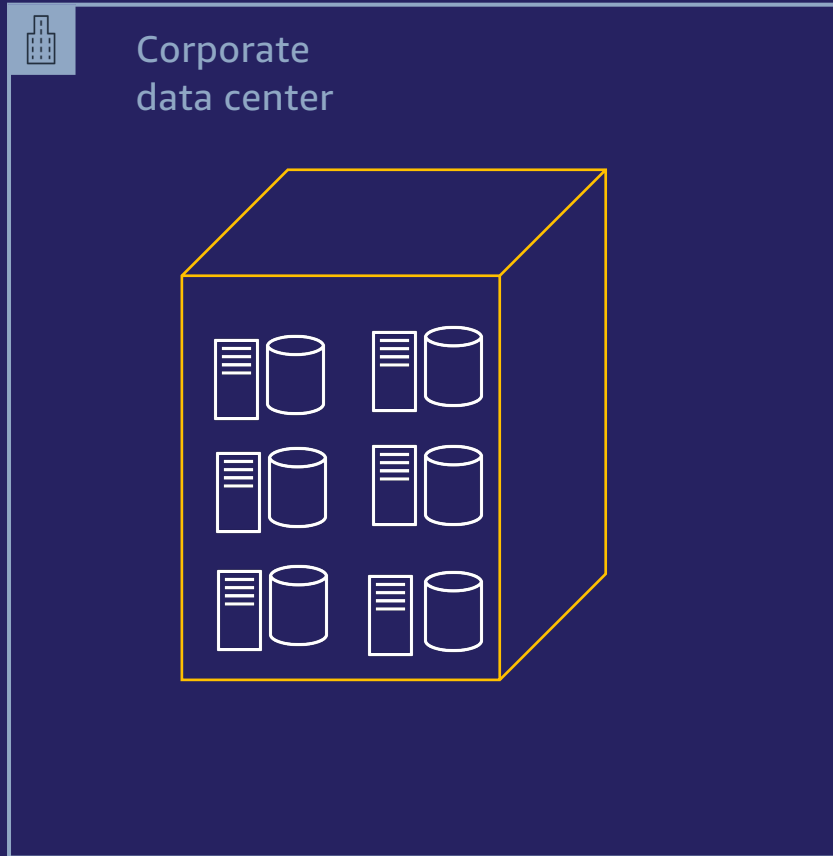
- コスト削減
- 生産性
- 対障害生
- 敏捷性



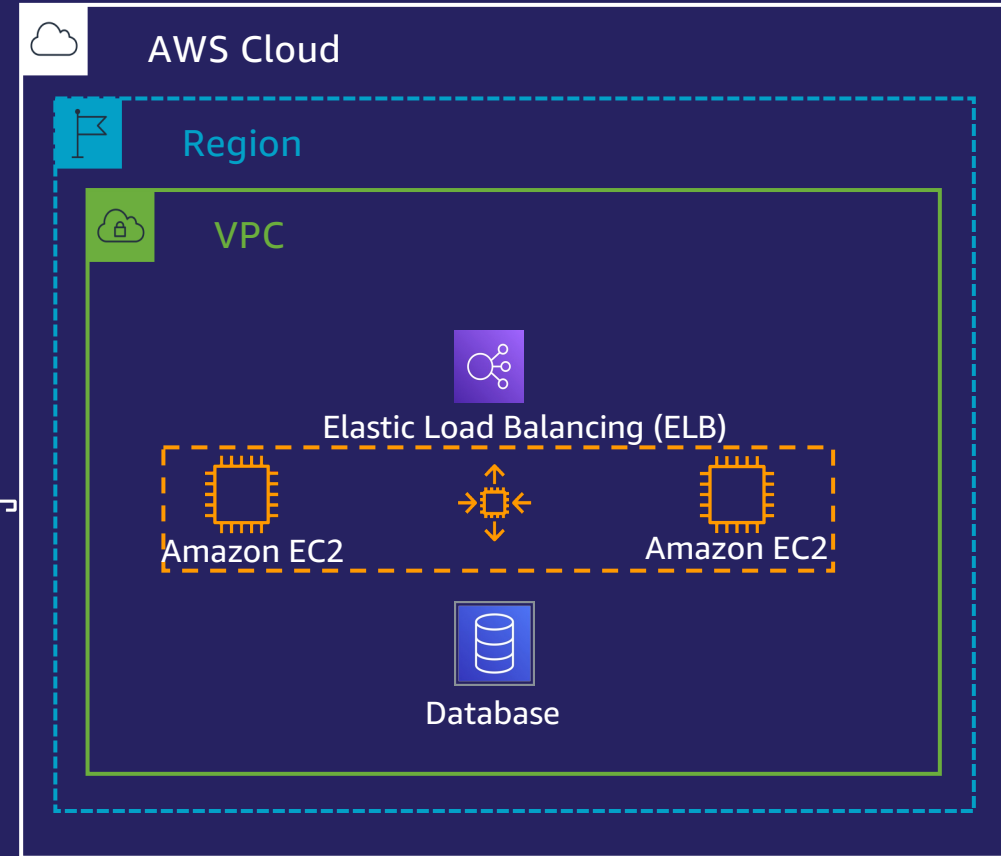
## フェーズ2 modernize(リファクタ)

AWSに移行後に、モダナイズ  
クラウドの利点を活かして容易に  
リファクタリング

# フェーズ1 : Lift and shift(Rehost)

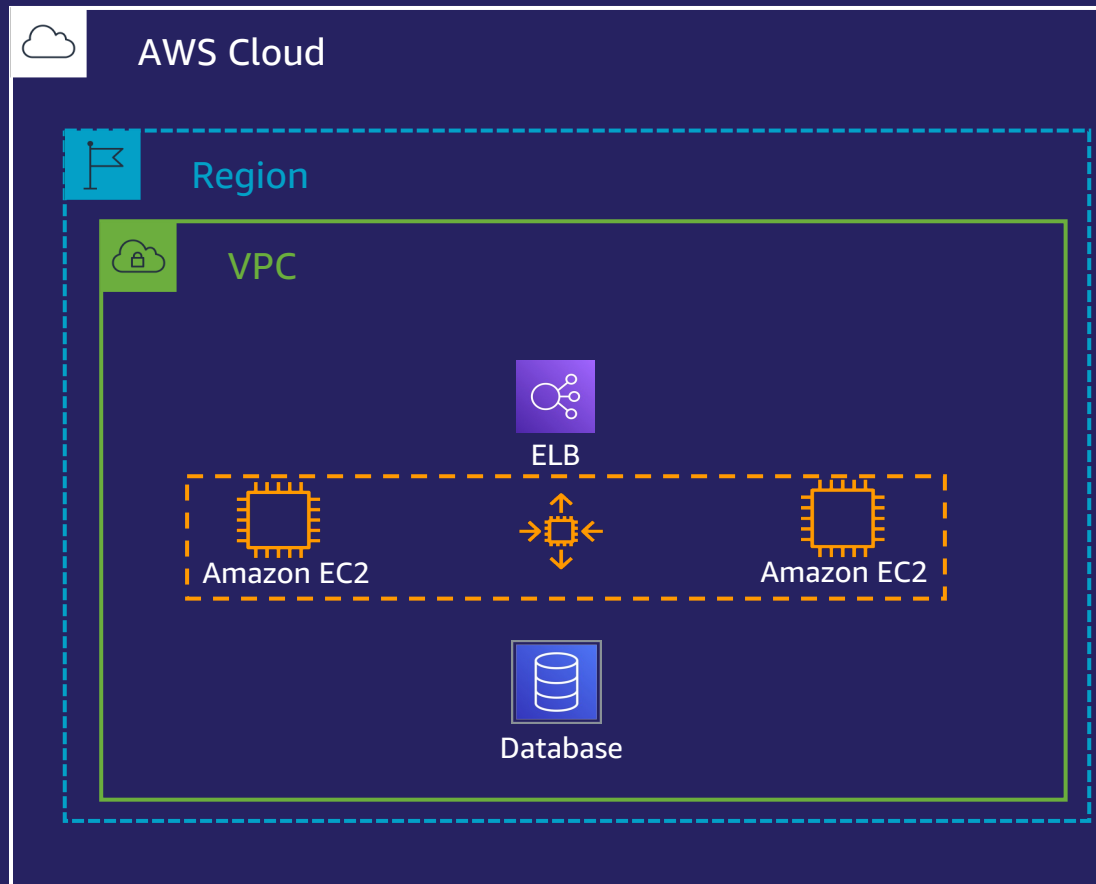


モノリス



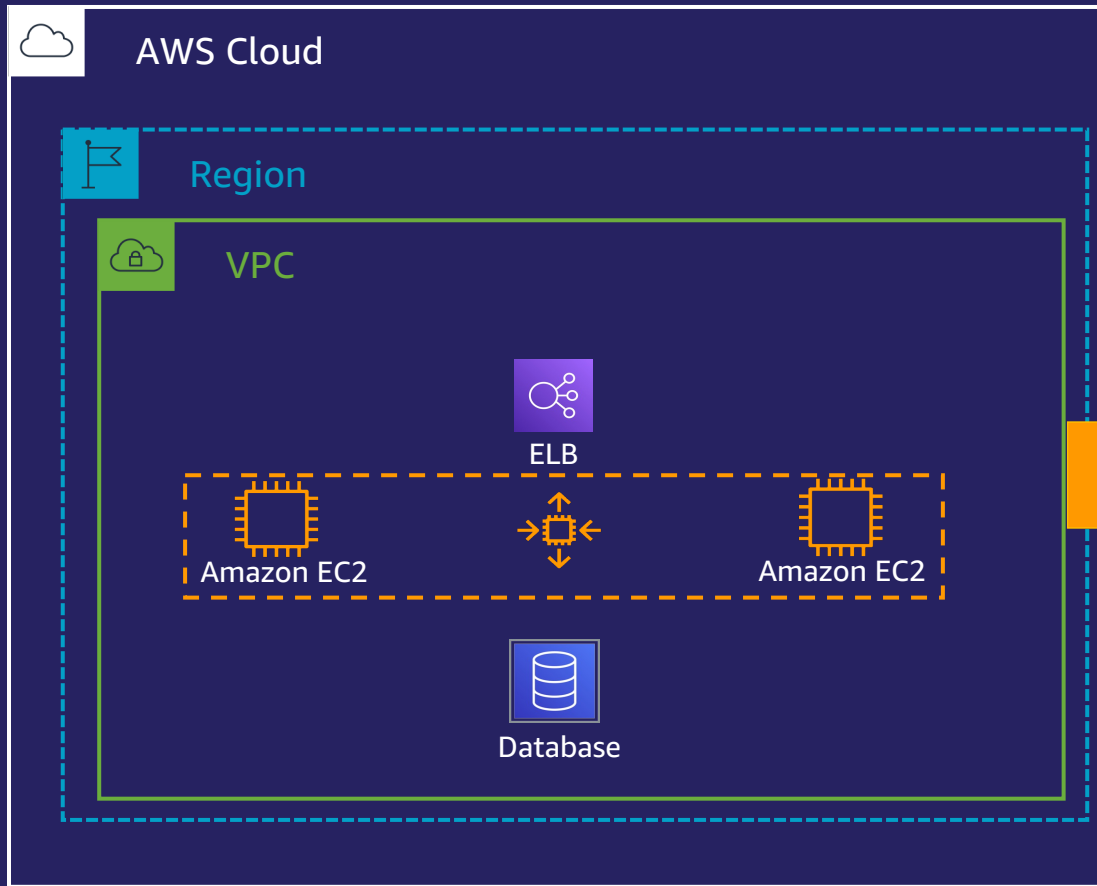
モノリス

# フェーズ2: Modernize (Refactoring)

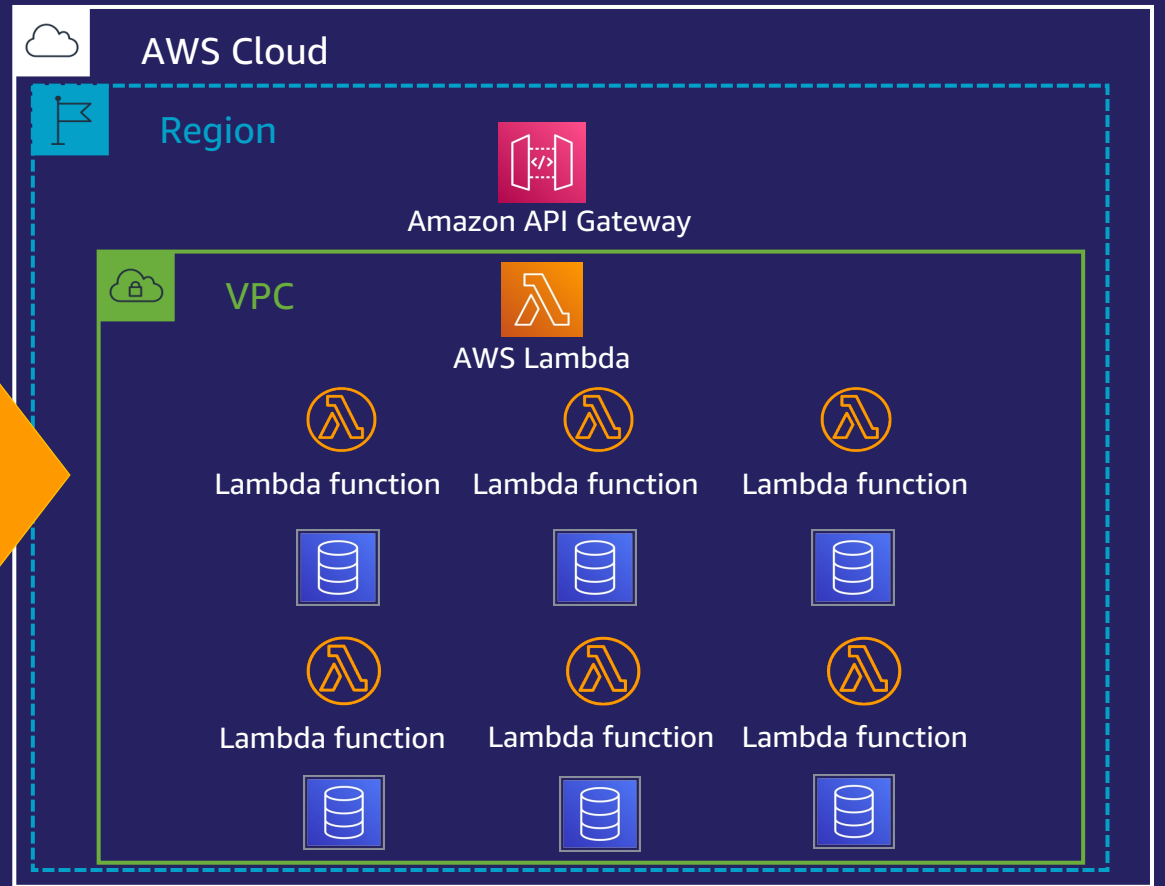


モノリス

# フェーズ2: Modernize (Refactoring)



モノリス



Refactor

# Strangler Fig パターンによるモダナイズ



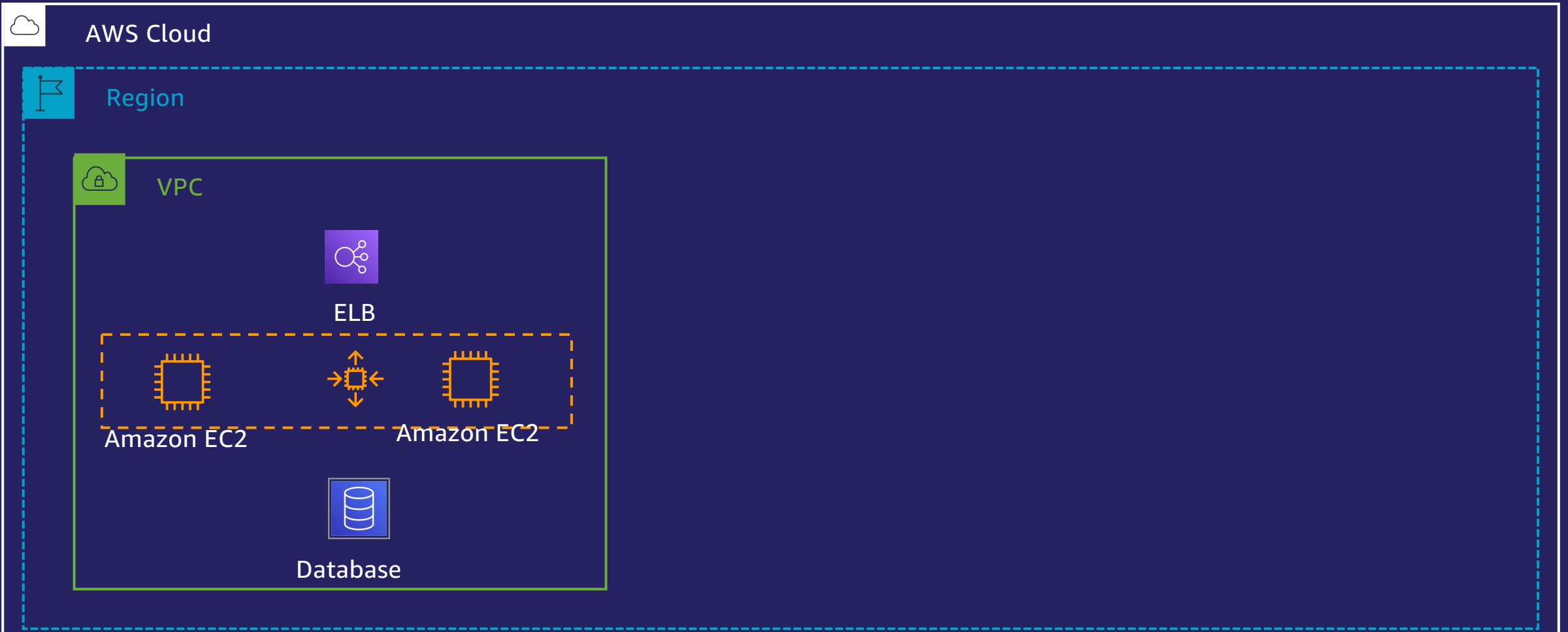
“Gradually create a new system around the edges of the old, letting it grow slowly over several years until the old system is strangled.”

Martin Fowler, Thoughtworks

## 利点

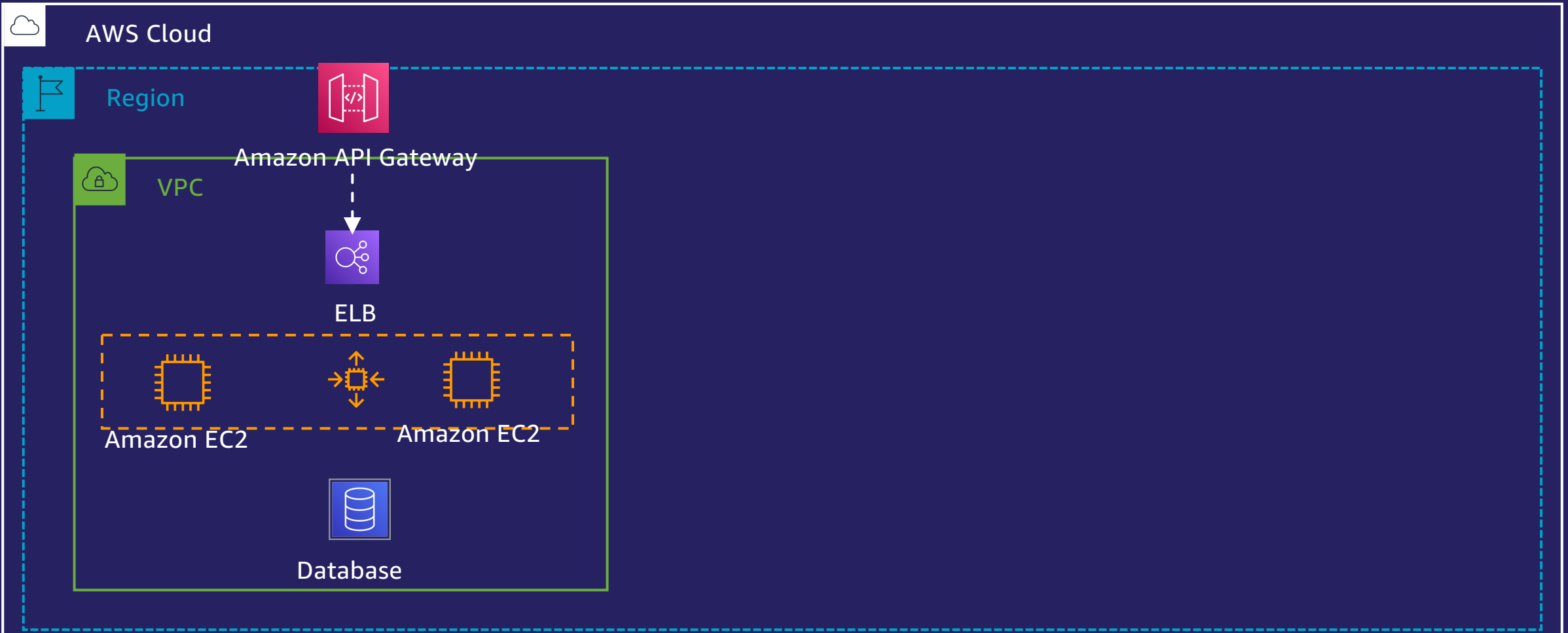
- 小さく俊敏に始められる
- すぐにモダナイズを始められる
- 容易に切り戻せてリスクを低減

# API Gatewayを使ったStrangler Fig パターン

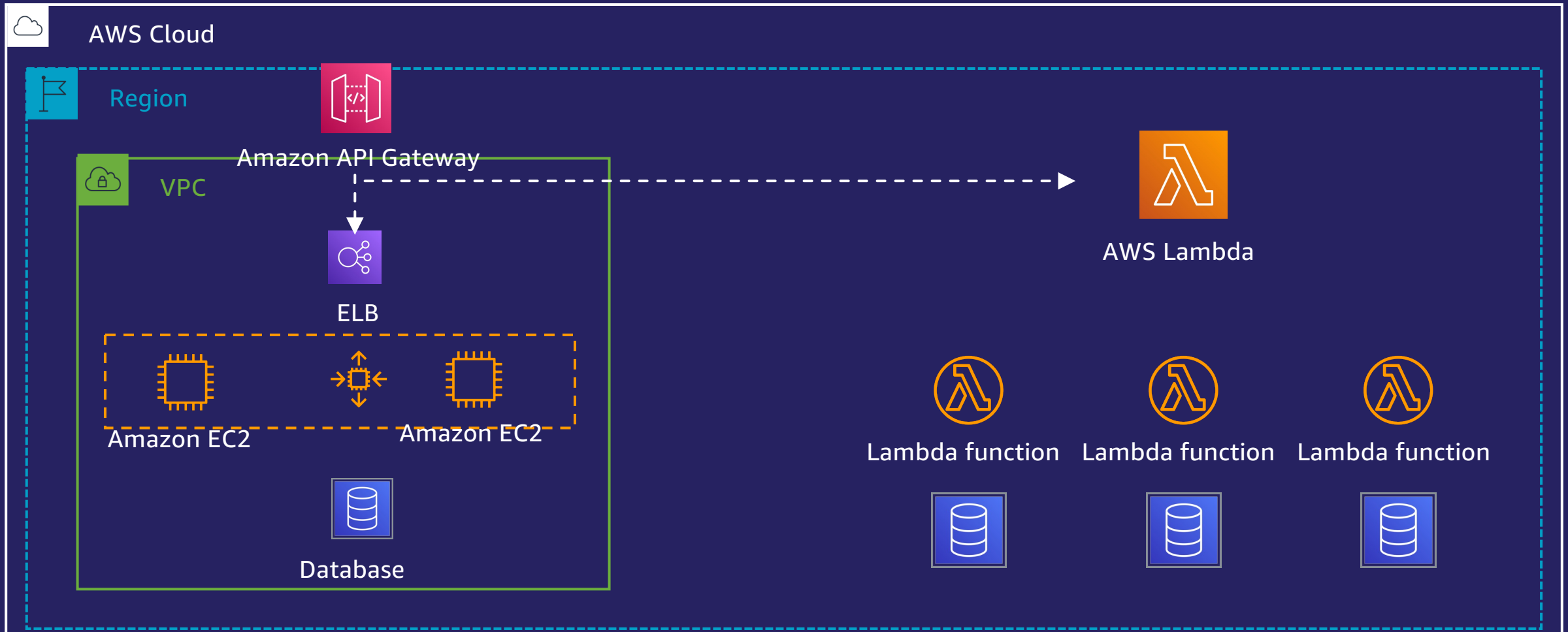




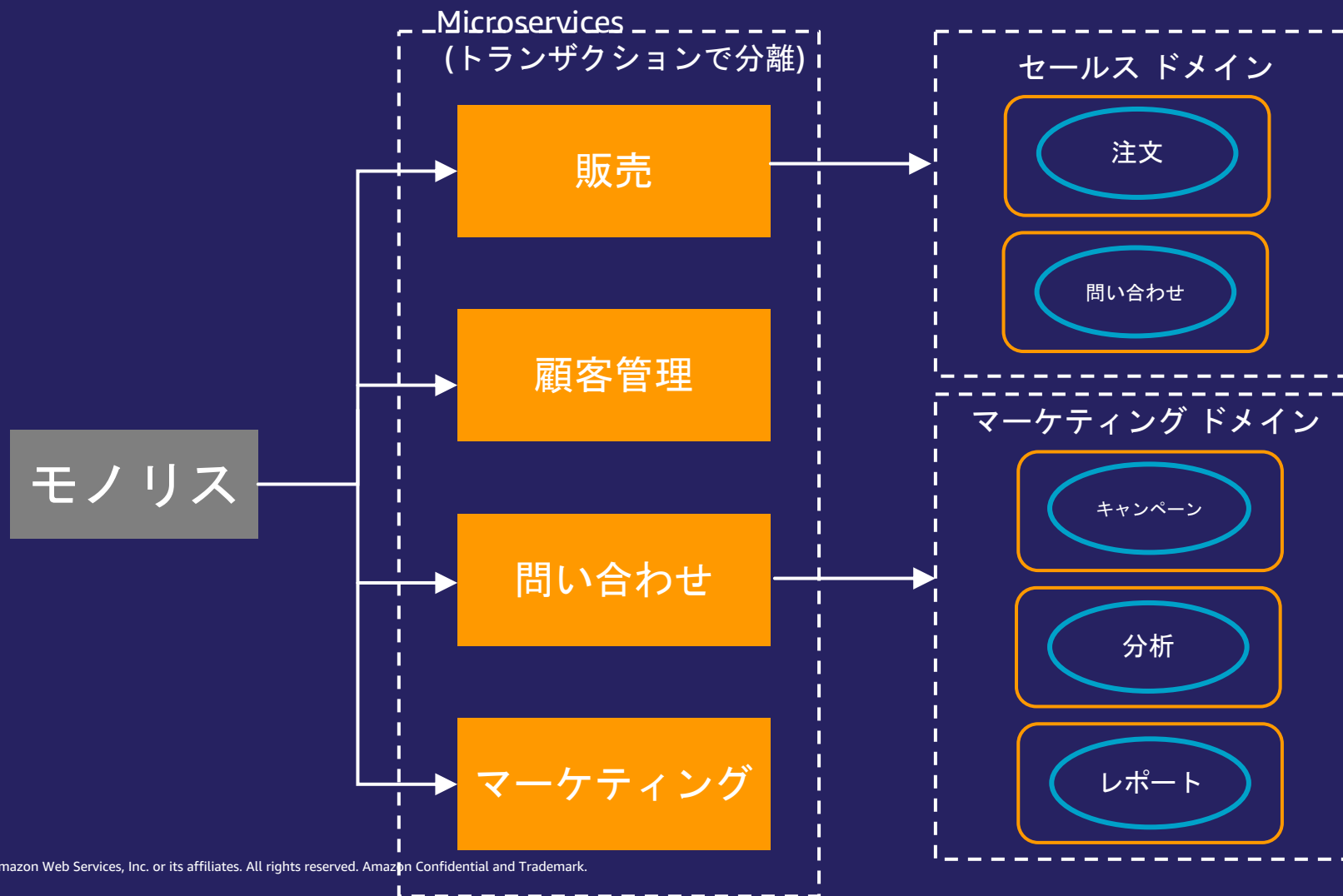
# API Gatewayを使ったStrangler Fig パターン



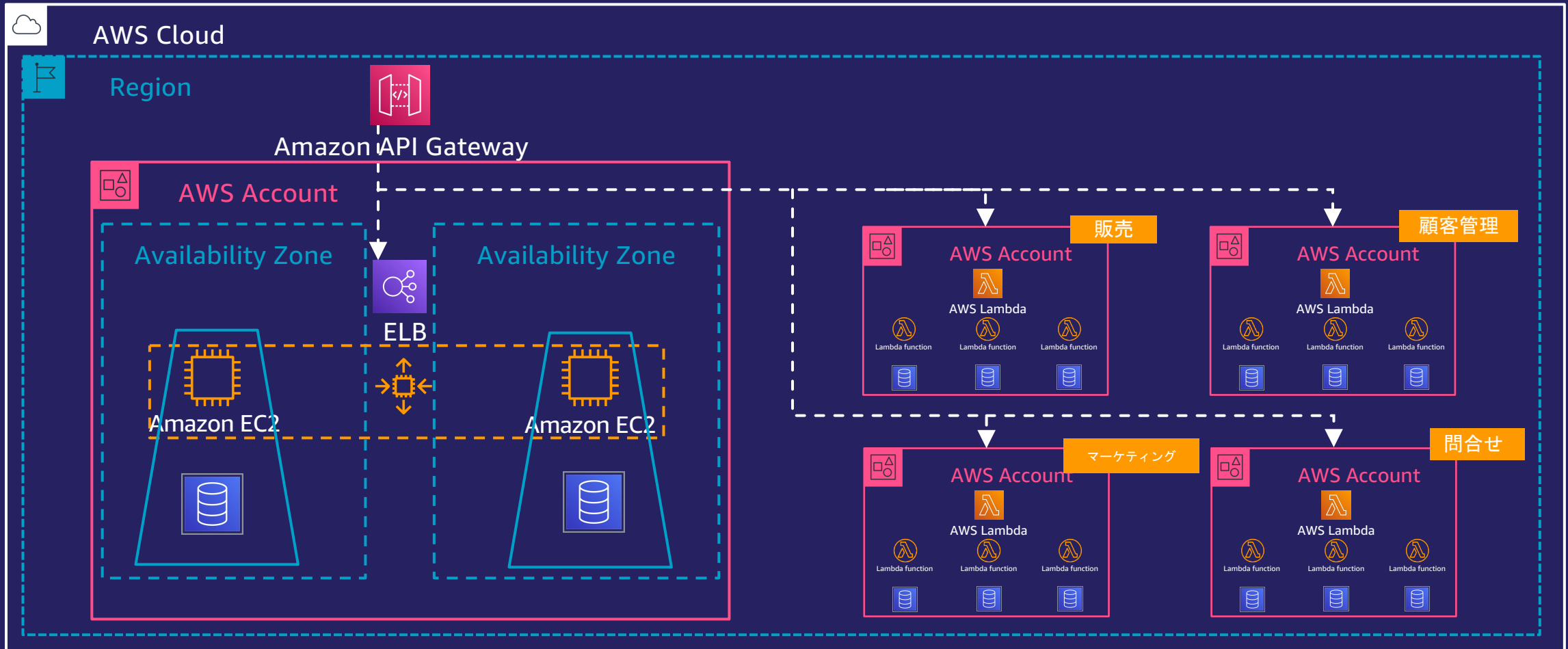
# API Gatewayを使ったStrangler Fig パターン



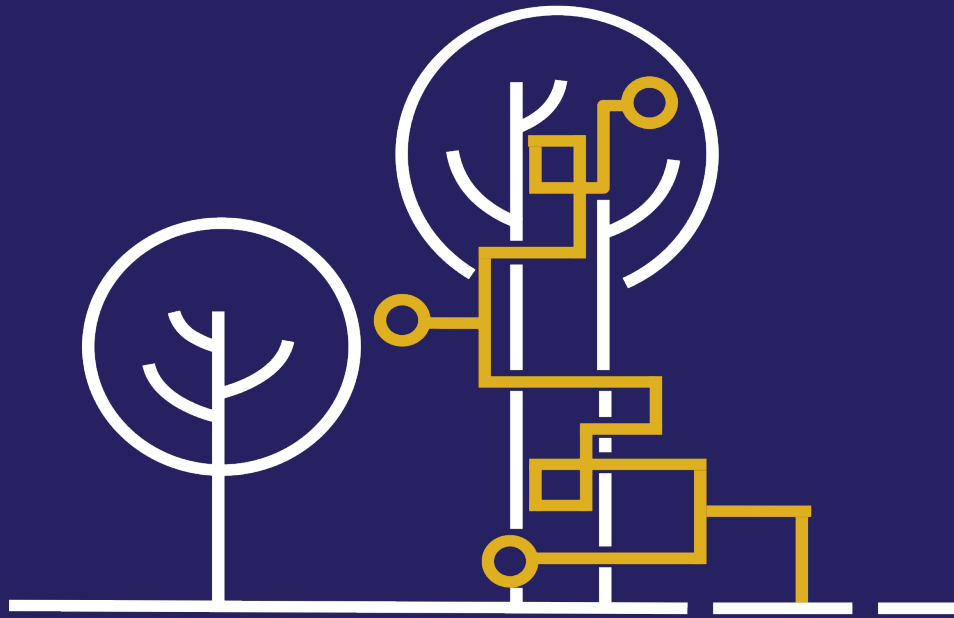
# 業務ドメインで分割



# 複数アカウントに分離すると複雑度が増加



# AWS Migration Hub Refactor Spaces



リファクタリングを  
数ヶ月から数日に短縮

単一アプリケーションを、容易に  
レガシーアプリケーションとマイクロサービスの組み合わせへ

# AWS Migration Hub Refactor Spacesの特徴



## AWS Migration Hub Refactor Spaces



リファクタリング環境  
のセットアップ・運用  
管理の手間を削減

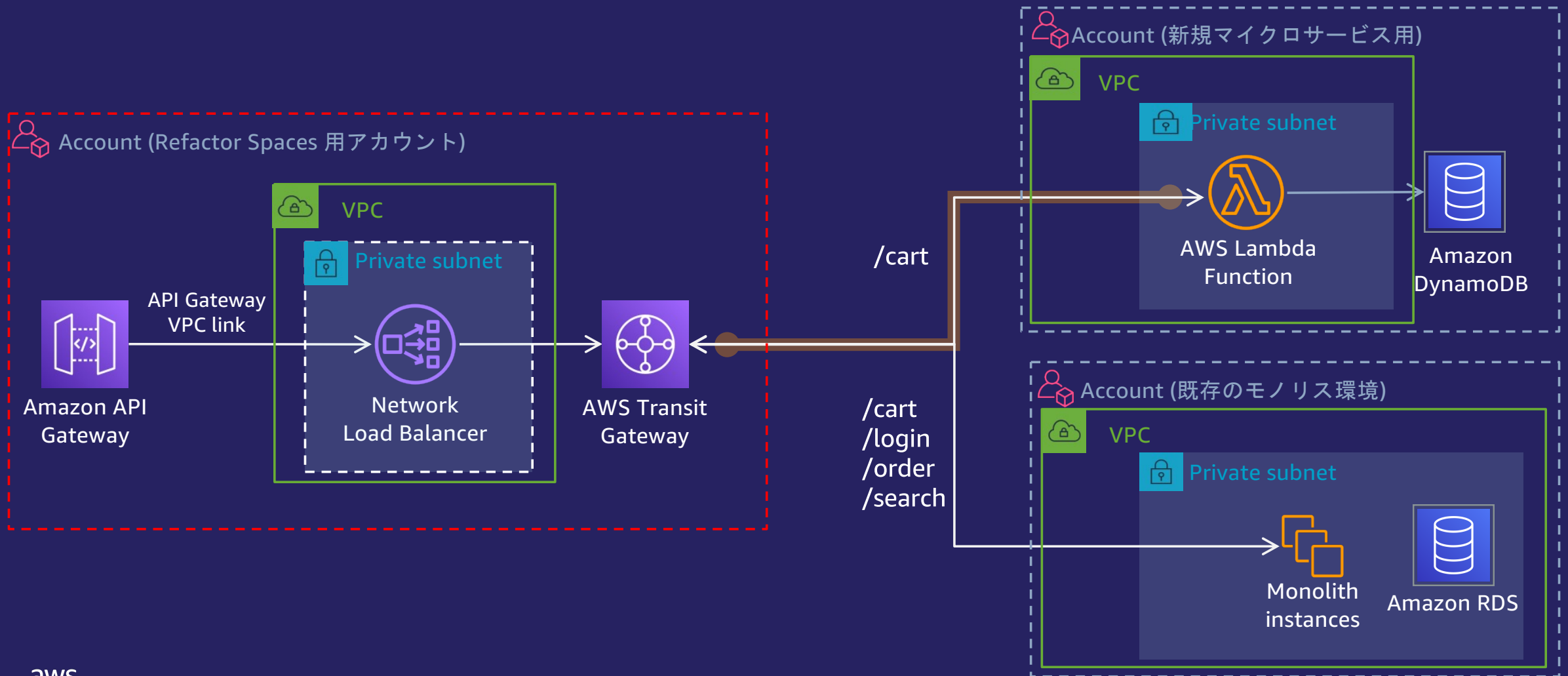


アプリケーションの呼  
び出し元に、リファク  
タリングの影響を与え  
ない



複数アカウントにまた  
がってトラフィックを  
ルーティング

# AWS Migration Hub Refactor Spacesの構成例



# AWS Migration Hub Refactor Spacesのまとめ

- モノリスを一気にリファクタリングするのは困難
- Strangler Figパターンで徐々にマイクロサービス化するのが一般的
- Strangler Figパターンを導入しやすくするためにアプリケーションプロキシ部分を構築するサービス





**Thank you!**