

AWS-06

企業内 Windows Server アプリケーション移行手法とツール

畠 泰三

技術統括本部 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



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

自己紹介

畠 泰三

所属：ソリューションアーキテクト

担当：通信業界のお客様に対しての技術支援

好きなAWSサービス：



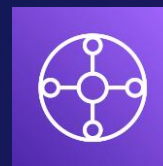
Amazon Elastic Compute
Cloud (Amazon EC2)



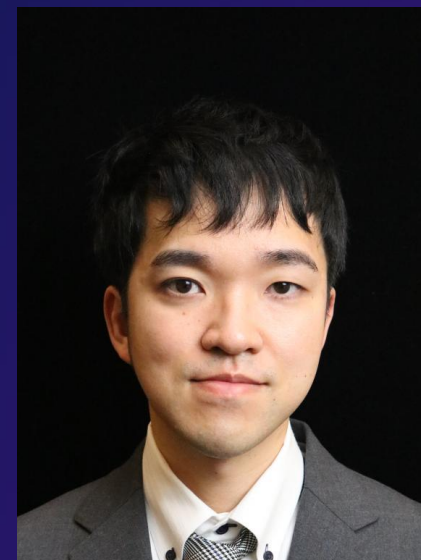
Amazon Elastic Container
Service (Amazon ECS)



AWS Lambda



AWS Transit Gateway



本セッションについて

対象者

- .NET Framework アプリケーション開発者の方
- .NET Framework アプリケーションや Microsoft SQL Server のクラウド移行をご検討されている方

ゴール

- .NET Framework アプリケーションの移行・モダナイゼーションシナリオについて理解する
- .NET Framework アプリケーション、Microsoft SQL Server の移行・モダナイゼーションに利用できるAWSサービス・ツールを知っていただく

本セッションで話さない事

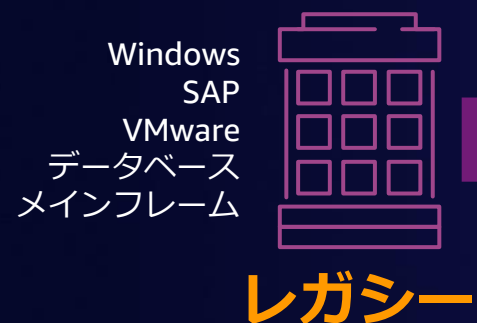
- 本セッションで紹介する個別の AWS サービス・ツールの詳細

アジェンダ






- 企業アプリケーションのクラウド移行
- .NET アプリケーションの移行シナリオ
- AWS が提供する モダナイゼーション ツール・サービス
- まとめ

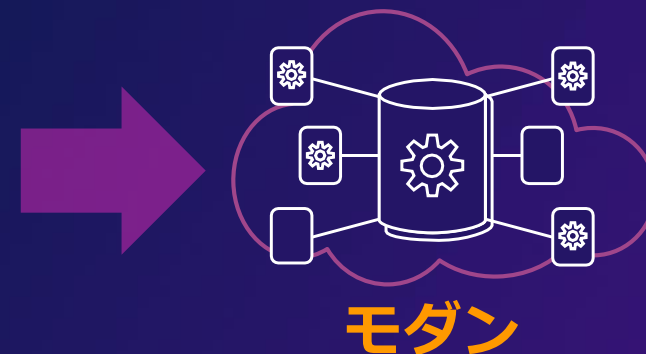
企業アプリケーションの クラウド移行

レガシーアプリケーションやデータはクラウドに移行し始めている



ゴール

-  ビジネス変革を加速
-  俊敏性を高め素早いイノベーションを起こす
-  運用コストの削減
-  クラウドの伸縮性、パフォーマンス、自動化
-  セキュリティとコンプライアンスの向上



>50%

の企業のワークロードとデータが 12か月以内に
パブリッククラウドに移行すると見込まれている

AWS のツール、プログラム、サービス



✓
目的に特化した
AWS および
パートナー製品

✓
インセンティブ
プログラム

✓
サードパーティー
オファリング

✓
無料の
リソース

✓
AWS
プロフェッショナル
サービス

✓
認定パートナー
サービス

AWS のイノベーションと経験



スケール



信頼性



セキュリティ



アジャイル
アプリケーション開発



パフォーマンス



高度なデータ分析

10 年以上

数千の移行とモダナイゼーションプロジェクト実績

モダナイゼーションとは

アプリケーションやインフラを変革していく中で、クラウドネイティブサービスの活用を幅を広げ、技術的負債の軽減や新しいビジネスの開拓・イノベーション促進等の**ビジネスの俊敏性を最大化**すること



レガシーアプリケーションが直面している課題



拡張性の欠如

急激なトラフィックの増加や
ユーザからの要望への対応



技術的負債

レガシーアーキテクチャの構築と維持
難易度の上昇



ライセンス費

予測不能なソフトウェアコストの増加



セキュリティ/コンプライアンス

古いソフトウェアによるセキュリティ
及びコンプライアンスリスク



イノベーションへの圧力

イノベーションの加速や高速な開発への
プレッシャー



求められるスキルの変化

オンプレミスと異なるクラウドの運用
モデル

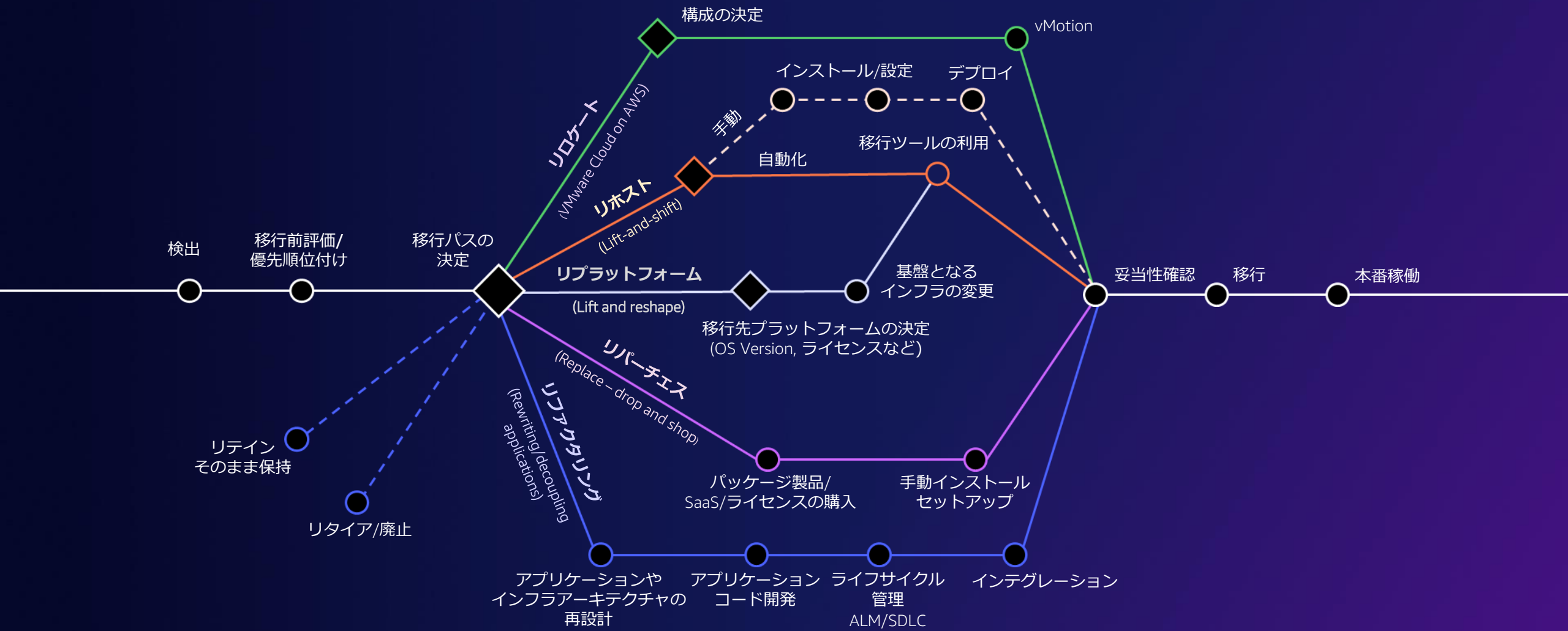
91%

COVID-19の影響によってクラウド利用
量が事前計画を上回ると予想

9.1%

全世界のIT投資におけるクラウド投資の
割合

AWS の移行戦略(7R)



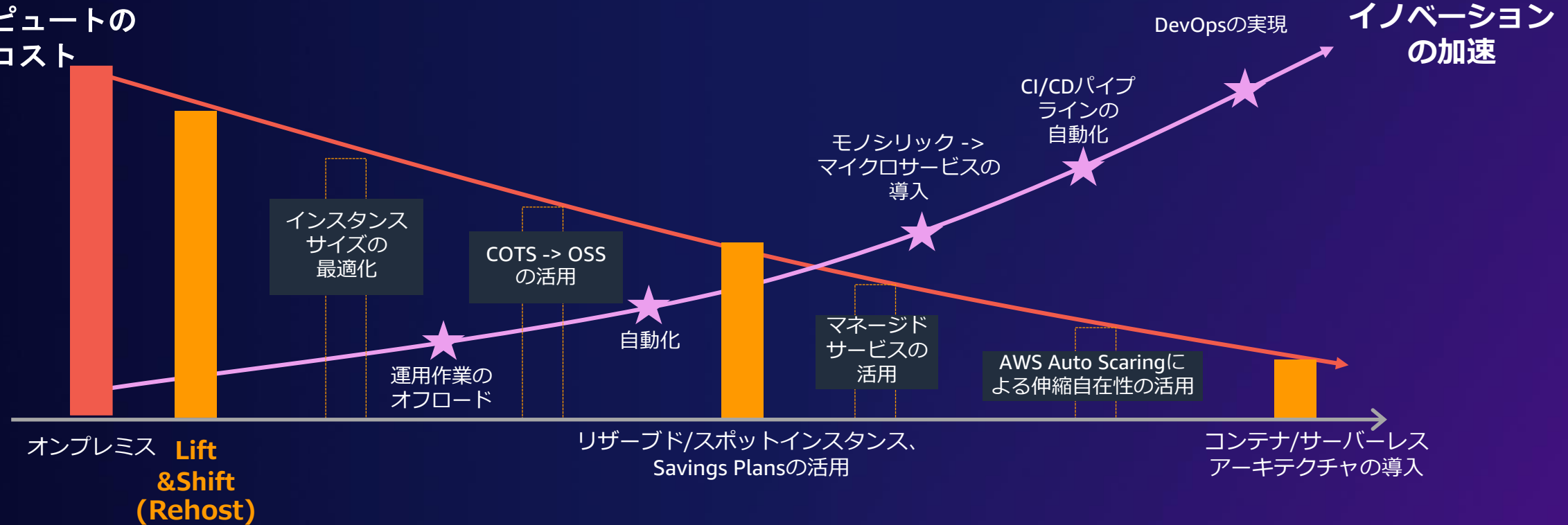
Rehost (Lift & Shift)とモダナイゼーション

レガシー

マイグレーション(移行)

モダナイゼーション

コンピュータの
コスト

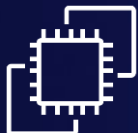


マイクロソフトウェアクロードのモダナイゼーション

Rehost

(Lift and Shift)

マイグレーション(移行)



Amazon EC2 Windows



SQL Server
on
Amazon EC2 Windows

Replatform

(Lift and Reshape)

コンテナ化



Amazon ECS



AWS Fargate



Amazon Elastic Kubernetes Service (Amazon EKS)

マネージドサービスの活用



Amazon Relational Database Service
(Amazon RDS)
for SQL Server

Refactor

(Rewrite/アプリケーションの疎結合化)

.NET 5+

.NET Core

サーバーレス化



AWS Lambda

クラウドネイティブデータベースの活用



Amazon Aurora



Amazon Neptune



Amazon DynamoDB



Amazon Redshift

Windows
アプリケーション

データベース

.NET アプリケーションの 移行シナリオ

“.NET Core is the Future of .NET”* – 変貌する.NET

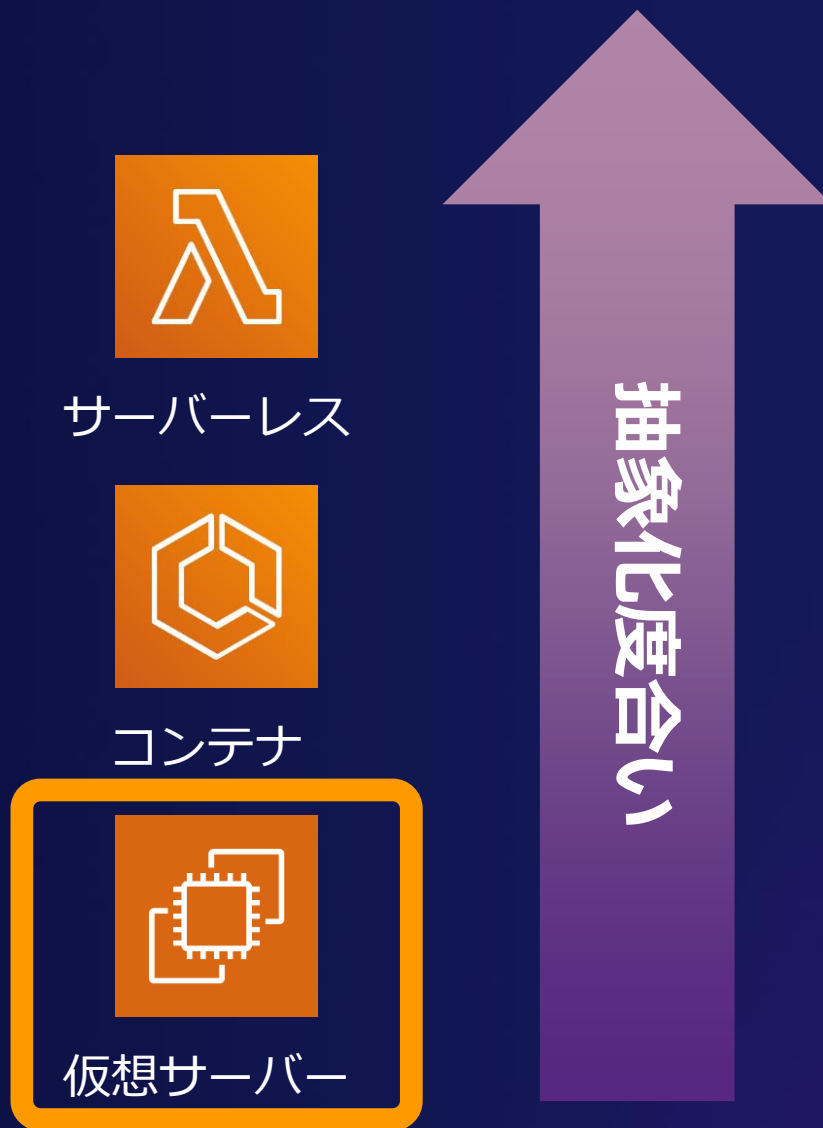
- “.NET Framework 4.8 は最後のメジャーバージョン”
- “...すべての新しい .NET アプリケーションは .NET Coreベースとなるべき”
- “今後、全ての .NET 投資は .NET Core に対して行われる”
 - Scott Hunter, Director Program Management for .NET, Microsoft*

<https://devblogs.microsoft.com/dotnet/net-core-is-the-future-of-net/> - May 6th, 2019

.NET Core リリーススケジュール計画と名称

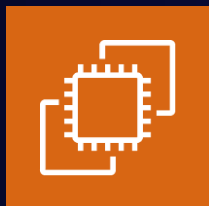
<u>.NET Core 3.1</u>	2019年12月	<u>現行バージョン、Long-Term Support (LTS)</u>
.NET 5.0	2020年11月	メンテナンスバージョンアップ
<u>.NET 6.0</u>	2021年11月	<u>Long-Term Support (LTS)</u>
.NET 7.0	2022年11月	メンテナンスバージョンアップ

.NET コードを走らせるコンピューティングリソース



.NET コードを走らせるコンピューティングリソース

抽象化度合い



仮想サーバー: Windows, Linux (.NET Framework, .NET Core)

- インスタンスサイズ、OSレイヤーなどの管理の**自由度が高い**

.NET コードを走らせるコンピューティングリソース



コンテナ: Windows*, Linux (.NET Framework, .NET Core)

- 自由度と抽象化度合いを両立
- 不変的かつ可搬性に優れる
- 高速なデプロイとスケーリングが容易

抽象化度合い

.NET コードを走らせるコンピューティングリソース



サーバーレス: Lambda (.NET Core)

- **抽象化度合いが非常に高い**短期実行型サーバーレスプラットフォーム (< 15分以下)
- コードが実行された時間での課金
- 豊富な言語選択肢 (C#, Node.js, Python, Java, Go, Ruby) と自己開発ランタイムの持ち込み

抽象化度合い

.NET コードを走らせるコンピューティングリソース



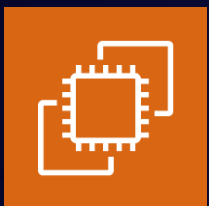
サーバーレス: Lambda (.NET Core)

- **抽象化度合いが非常に高い**短期実行型サーバーレスプラットフォーム (< 15分以下)
- コードが実行された時間での課金
- 豊富な言語選択肢 (C#, Node.js, Python, Java, Go, Ruby) と自己開発ランタイムの持ち込み



コンテナ: Windows*, Linux (.NET Framework, .NET Core)

- **自由度と抽象化度合いを両立**
- 不変的かつ可搬性に優れる
- 高速なデプロイとスケーリングが容易

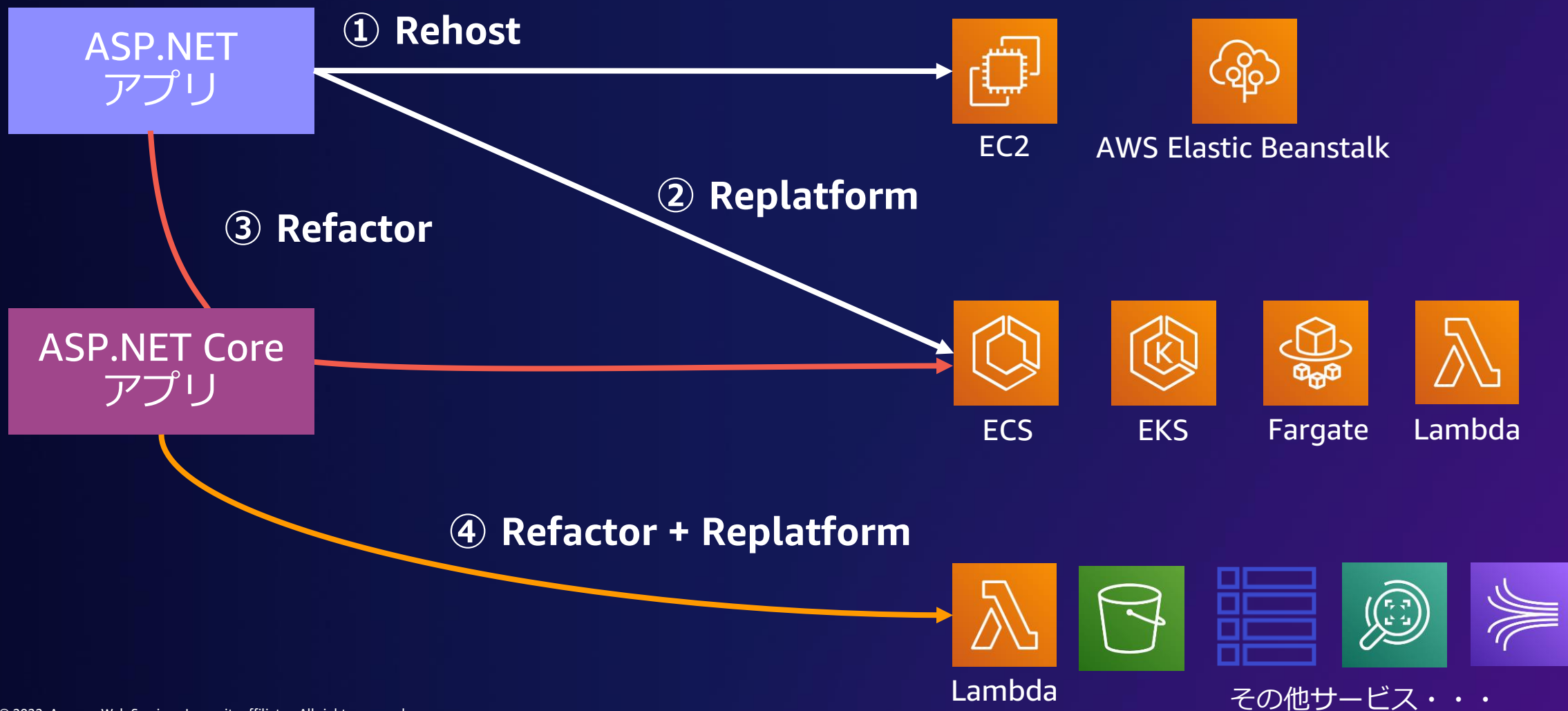


仮想サーバー: Windows, Linux (.NET Framework, .NET Core)

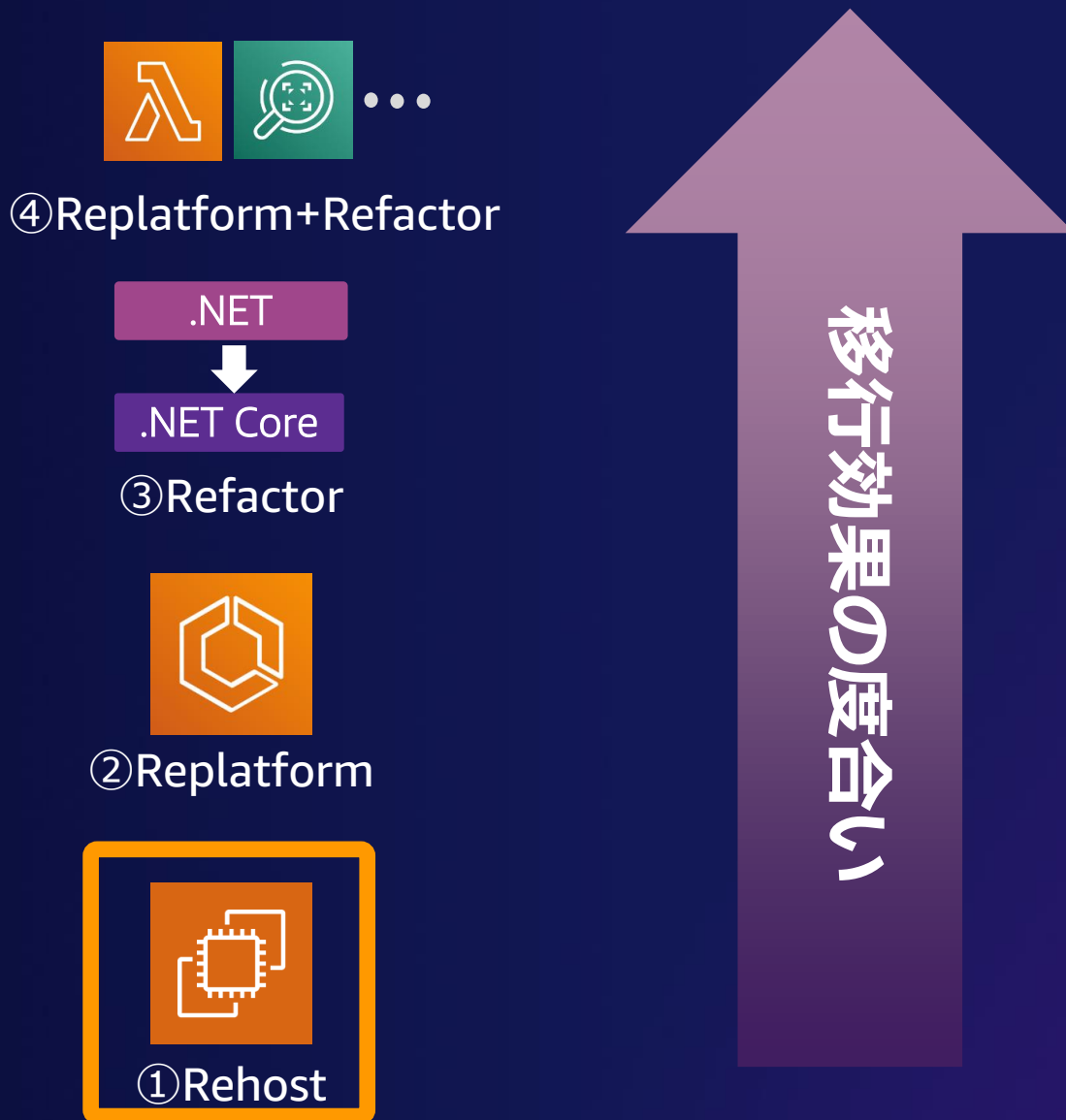
- インスタンスサイズ、OSレイヤーなどの**管理の自由度が高い**

抽象化度合い

ASP.NET アプリケーション における 7R

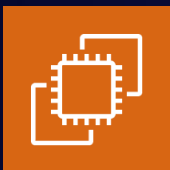


.NET Framework マイグレーション戦略



.NET Framework マイグレーション戦略

移行効果の度合い



①Rehost

- コード変更をほぼ行わず、クラウドメリットを獲得
- IIS, .NET Framework などの依存関係を柔軟に選択
 - × .NET Framework に対する機能追加が行われない

.NET Framework マイグレーション戦略

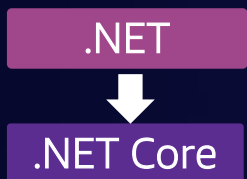


②Replatform

- 最小限のコード変更で、サーバー集約と CI/CD (DevOps) を実現
- 異なる .NET Framework バージョンと IIS の組み合わせをサポート
 - × .NET Framework 自体の課題は残る

移行効果の度合い

.NET Framework マイグレーション戦略



③Refactor

- **.NET Core on Linux へ移行**し、ライセンスコストの低減
- 従量課金をフルに活用したランニングコストの効率化

移行効果の度合い

.NET Framework マイグレーション戦略



④Replatform+Refactor

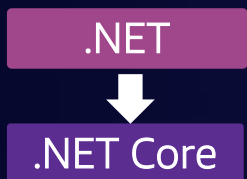
- アプリケーションアーキテクチャ自体を大幅に刷新
- マネージドサービスをフル活用し、継続的刷新サイクルへ

移行効果の度合い

.NET Framework マイグレーション戦略



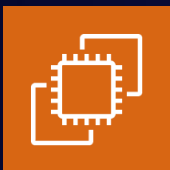
④Replatform+Refactor



③Refactor



②Replatform



①Rehost

- アプリケーションアーキテクチャ自体を大幅に刷新
- **マネージドサービスをフル活用**し、継続的刷新サイクルへ
- **.NET Core on Linux へ移行**し、ライセンスコストの低減
- 従量課金をフルに活用したランニングコストの効率化
- **最小限のコード変更で、サーバー集約と CI/CD (DevOps) を実現**
- 異なる .NET Framework バージョンと IIS の組み合わせをサポート
 - × .NET Framework 自体の課題は残る
- **コード変更をほぼ行わず、クラウドメリットを獲得**
- IIS, .NET Framework などの依存関係を柔軟に選択
 - × .NET Framework に対する機能追加が行われない

移行効果の度合い

ASP.NET アプリケーションのマイグレーション

まず確認したいこと → 使用しているサブフレームワークの種類

サブフレームワークの例

ASP.NET
Windows 環境
でのみ動作

- ASP.NET Web Forms
- ASP.NET MVC
- ASP.NET Web API
- ASP.NET SignalR

- マークアップなしの簡単 Web アプリ開発
- MVCフレームワークでの本格 Web アプリ開発
- Web API 開発
- WebSocket を使った双方向通信

ASP.NET Core
クロスプラット
フォーム
Windows, macOS,
Linuxで動作

- ASP.NET Core MVC
- ASP.NET Core WebAPI
- ASP.NET Core SignalR
- ASP.NET Core Razor Pages

- MVCフレームワークでの本格 Web アプリ開発
- Web API 開発
- WebSocket を使った双方向通信
- ページコーディングに集中した開発

ASP.NET(Core)アプリケーションのマイグレーション先 AWS サービス

○：移行できる ×：移行できない

		EC2	Elastic Beanstalk	ECS on EC2	EKS on EC2	ECS on Fargate	EKS on Fargate*	Lambda*
ASP.NET	Web Forms	○	○	○	○	○	×	×
	MVC	○	○	○	○	○	×	×
	Web API	○	○	○	○	○	×	×
	SignalR	○	○	×	×	×	×	×
ASP.NET Core	MVC	○	○	○	○	○	○	○
	Web API	○	○	○	○	○	○	○
	Razor Pages	○	○	○	○	○	○	○
	SignalR	○	○	×	×	×	×	×

- EKS on Fargate は Windows コンテナ非対応のため ASP.NET アプリは不可
- Lambda は .NET Framework 非対応のため ASP.NET アプリは不可
- SignalR は WebSocket を使用してアプリがその接続管理を行うため、接続先ノードが変わる可能性のある ECS, EKS, Lambda では ASP.NET アプリを動かすのは不可と判断

AWS が提供する モダナイゼーション ツール・サービス

AWS が提供するモダナイゼーションサービスやツール

Application Modernization

アセスメント

Migration Hub Strategy Recommendations

規模に応じたアセスメント



Migration + Modernization
Assessment+
Recommendations

Replatform

AWS App2Container

モダナイゼーションの加速



EC2 Windows
➤ Win Containers

Refactor

Porting Assistant for .NET

手作業の最小化

.NET Core

.NET Framework
➤ .NET Core

AWS Microservice Extractor for .NET

リファクタリングの加速



Monolith
➤ Microservices

Database Modernization

AWS Database Migration Service

高速、安全なデータベース
マイグレーション



SQL Server
➤ Aurora or other

AWS Schema Conversion Tool

手作業の最小化



SQL Server
➤ Aurora or other

Replatforming Assistant

リプラットフォームの
自動化



SQL Server
➤ EC2 Linux

Babelfish for Aurora PostgreSQL

モダナイゼーションの加速



SQL Server
➤ Aurora

AWS App2Container

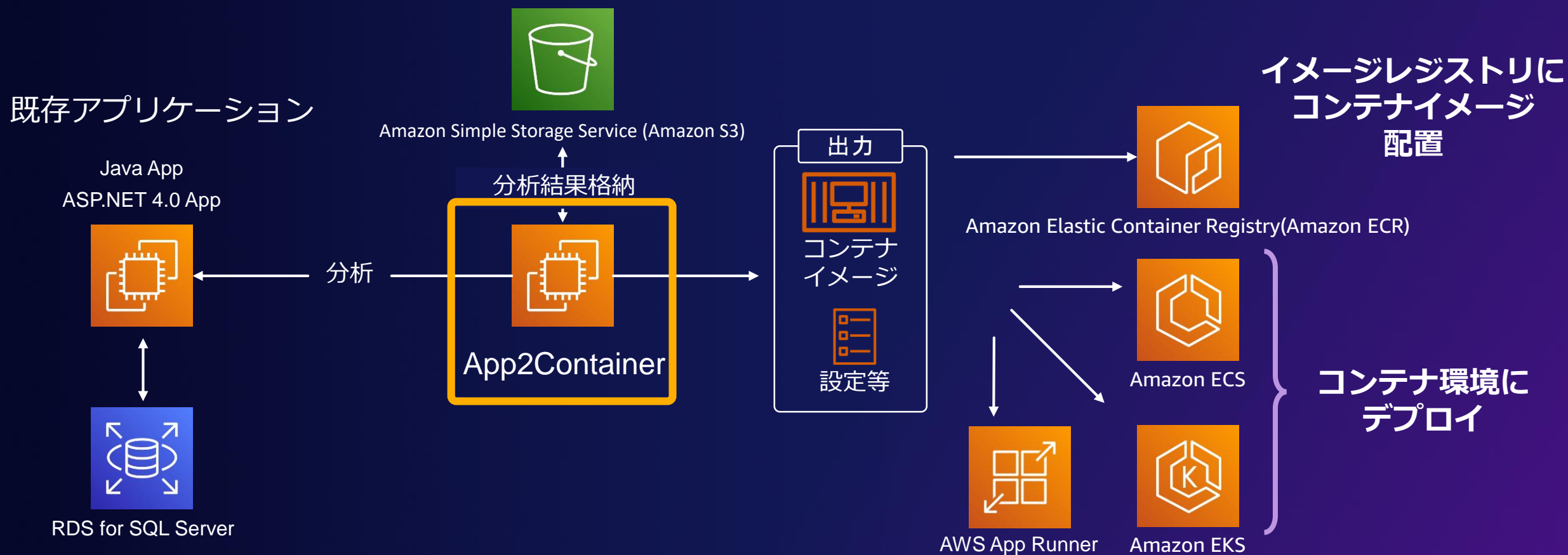
.NET 及び Java アプリケーションのコンテナ化により移行を支援

- .NET 及び Java アプリケーションのソースコードからコンテナイメージ等を自動生成

Replatform



EC2 Windows
➤ Win Containers



Porting Assistant for .NET

.NET Framework から .NET Core への移植に関するインサイトと支援

- 互換性調査や、単純に置換可能な部分のソースコード変換を行うツール

Refactor

.NET Core

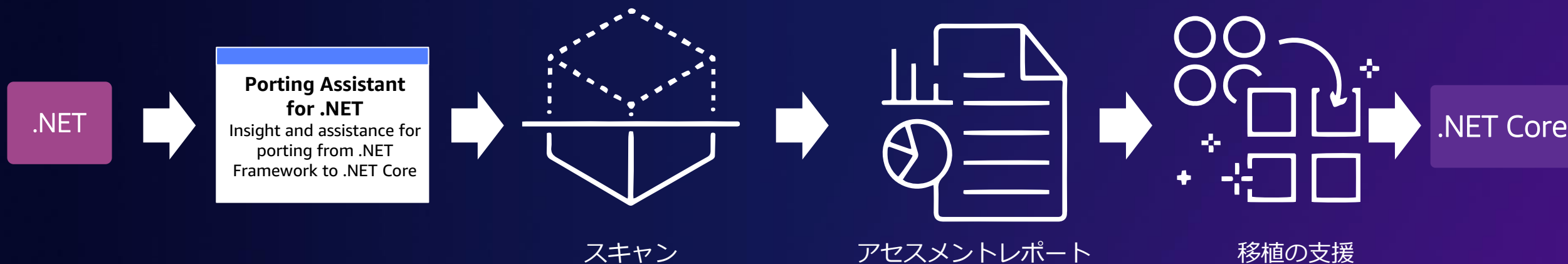
.NET Framework
➤ .NET Core

作業の優先順位付けに活用

.NET Core 互換性評価レポートを生成し、移植工数の算出や優先順位付けが簡単に

調査工数の削減に活用

互換性のない .NET Core API とパッケージを識別し、代替案の情報をご提示により互換性調査工数を削減



AWS Microservice Extractor for .NET

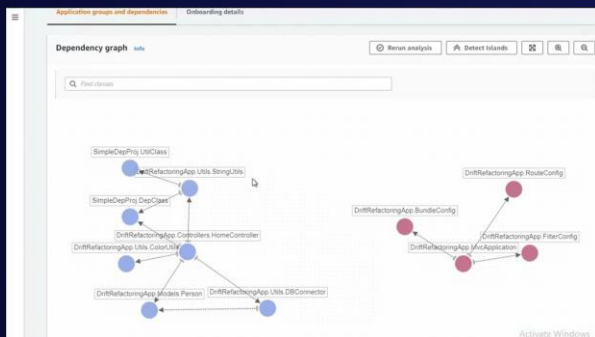
モノシリックな .NET アプリケーションのリファクタを支援

- .NET アプリケーションのマイクロサービスへのリファクタを支援
- ソースコードとランタイムメトリクスを分析し、アプリケーションとその依存関係を視覚化

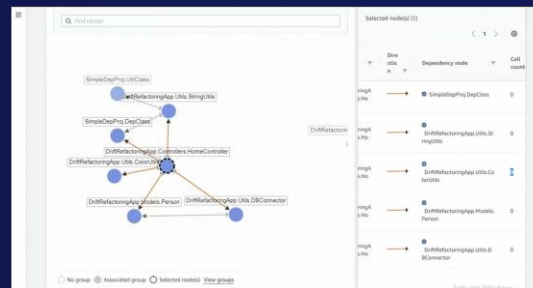
Refactor



Monolith
➤ Microservices



グラフ形式での視覚化



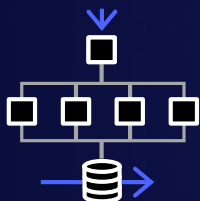
関連ランタイムの呼び出し数の視覚化

Direction	Dependency node	Call count
ringA.s.Ho	SimpleDepProj.DepClass	0
ringA.s.Ho	DriftRefactoringApp.Utils.StringUtils	0
ringA.s.Ho	DriftRefactoringApp.Utils.ColorUtils	0

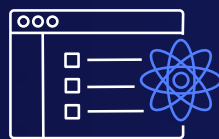
抽出・可視化プロセス



ソースコードとランタイムメトリクスから.NETアプリケーションをスキャン



依存関係をグラフ形式で視覚化



静的なコード解析結果として、ランタイム呼び出し数を単一ビューで視覚化



グラフを使って依存関係を理解し、グループとして整理するための構成要素ラベル付け



自動的にローカルAPIコールをREST APIコールへリファクタリングまた置換



アプリケーションの一部を個別のコードプロジェクトに抽出

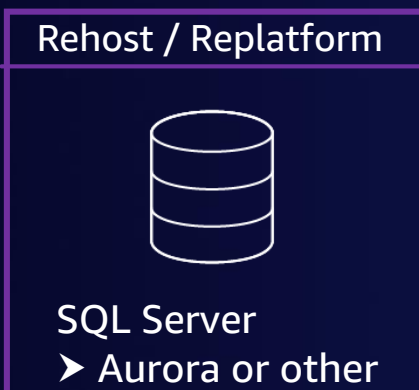


データベース モダナイゼーション サービス・ツール

AWS Database Migration Service

簡単かつ安全にデータベースや
データウェアハウスを AWS へ
マイグレーション

高速かつ安全にデータベースを
マイグレーション



Rehost / Refactor / Replatform



SQL Server

➤ on EC2 / RDS / Aurora

データベースモダナイゼーション サービス・ツール

Rehost / Refactor / Replatform



SQL Server
▶ on EC2 / RDS / Aurora

AWS Database Migration Service

簡単かつ安全にデータベースや
データウェアハウスを AWS へ
マイグレーション

高速かつ安全にデータベースを
マイグレーション

Rehost / Replatform



SQL Server
▶ Aurora or other

AWS Schema Conversion Tool

SQL Server のスキーマをターゲット
データベースのスキーマへのコンバートや
アセスメントレポートを作成

手作業の最小化

Replatform



SQL Server
▶ Aurora or other

データベースモダナイゼーション サービス・ツール

AWS Database Migration Service

簡単かつ安全にデータベースやデータウェアハウスを AWS へマイグレーション

高速かつ安全にデータベースを
マイグレーション

Rehost / Replatform



SQL Server
➤ Aurora or other

AWS Schema Conversion Tool

SQL Server のスキーマをターゲットデータベースのスキーマへのコンバートやアセスメントレポートを作成

手作業の最小化

Replatform



SQL Server
➤ Aurora or other

Replatforming Assistant

Windows から Linux への SQL Server リプラットフォームを自動化

Windowsライセンス費用の
削減

Replatform



SQL Server
➤ EC2 Linux

Rehost / Refactor / Replatform



SQL Server
➤ on EC2 / RDS / Aurora

Babelfish for Aurora PostgreSQL

SQL Server から Aurora PostgreSQL への移行リスク・工数・コストを削減

Refactor



SQL Server
➤ Aurora

- 最小限のコード変更または、コード変更を行わずに SQL Server 向けのアプリケーションを Aurora PostgreSQL で稼働

リスクの軽減



Translation-layer により、SQL Server の T-SQL を Aurora PostgreSQL に対して実行可能に

移行を加速



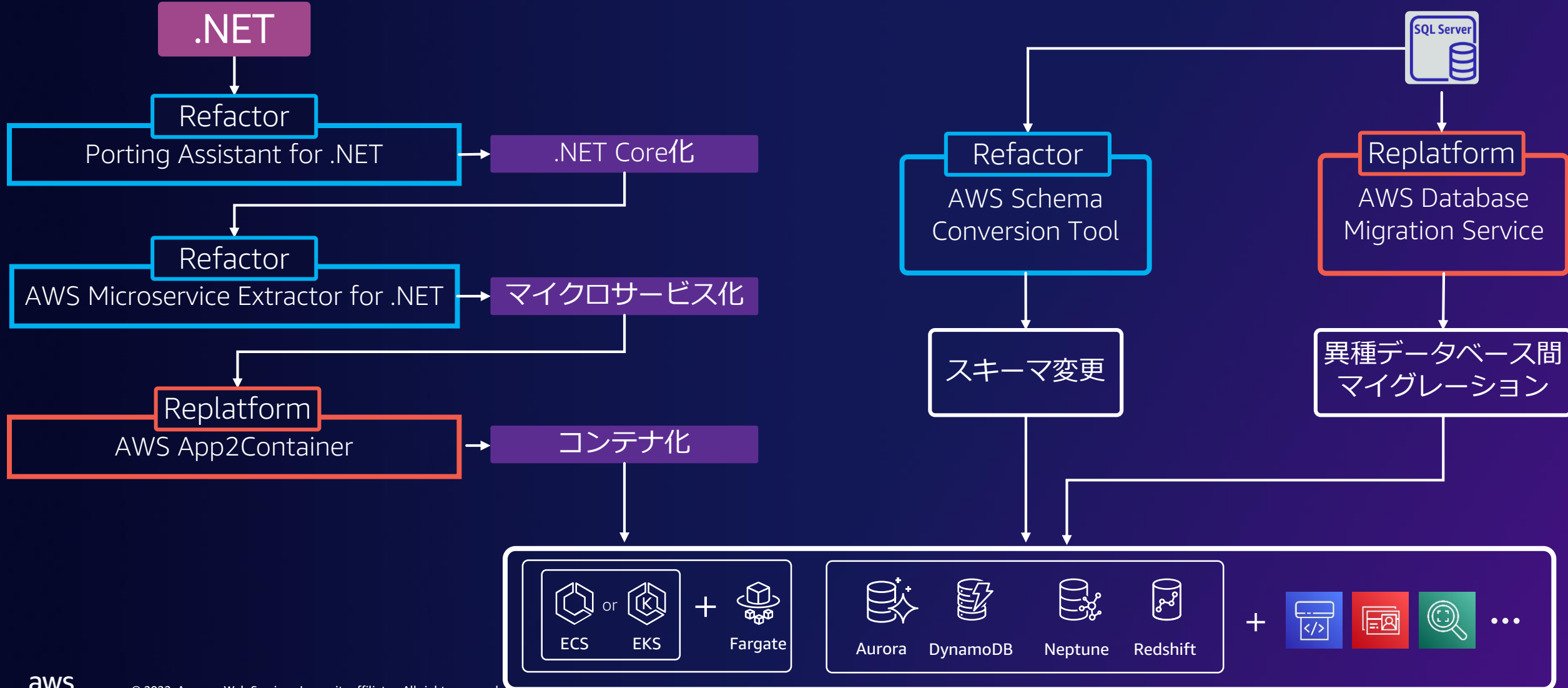
アプリケーションコードの改修工数を大幅に削減する事で SQL Server から Aurora への移行をより迅速に実現

Freedom to innovate



SQL Server のコードと 新しい PostgreSQL 用のコードを side-by-side で実行できるため、より慣れた環境で開発が可能

サービスやツールを組み合わせたモダナイゼーション



まとめ

まとめ

- ✓ 企業レガシーアプリケーションとデータのクラウド移行はトレンドに
- ✓ クラウド移行とモダナイゼーションによってコスト削減やビジネスの俊敏性を最大化・イノベーションの加速が可能
- ✓ 10年以上に渡る、数千のクラウドマイグレーションとモダナイゼーションの実績
- ✓ Windows Server アプリケーション や SQL Server のクラウドマイグレーション・モダナイゼーションに活用頂けるツールやマネージドサービスを提供

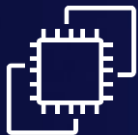
マイクロソフトウェアクロードのモダナイゼーション

Rehost
(Lift and Shift)

Replatform
(Lift and Reshape)

Refactor
(Rewrite/アプリケーションの疎結合化)

マイグレーション(移行)



EC2 Windows

コンテナ化



ECS



Fargate



EKS

.NET 5+

.NET Core

サーバーレス化



Lambda

Windows applications

SQL databases



SQL Server on EC2 Windows

マネージドサービスの活用



RDS for SQL Server

クラウドネイティブデータベースの活用



Aurora



DynamoDB



Neptune



Redshift

AWSが提供するモダナイゼーションサービスやツール

Application Modernization

アセスメント

Migration Hub Strategy Recommendations

規模に応じたアセスメント



Migration + Modernization
Assessment+
Recommendations

Replatform

AWS App2Container

モダナイゼーションの加速



EC2 Windows
➤ Win Containers

Refactor

Porting Assistant for .NET

手作業の最小化

.NET Core

.NET Framework
➤ .NET Core

AWS Microservice Extractor for .NET

リファクタリングの加速



Monolith
➤ Microservices

Database Modernization

AWS Database Migration Service

高速、安全なデータベース
マイグレーション



SQL Server
➤ EC2 Windows

AWS Schema Conversion Tool

手作業の最小化



SQL Server
➤ Aurora or other

Replatforming Assistant

リプラットフォームの
自動化



SQL Server
➤ EC2 Linux

Babelfish for Aurora PostgreSQL

モダナイゼーションの加速



SQL Server
➤ Aurora

マイグレーション関連セッション

- AWS-01** AWSクラウドマイグレーション成功への道クラウド移行
トータル支援プログラムのご紹介
- AWS-02** メインフレームからのクラウド移行事例と最新アップデート
- AWS-03** エンタープライズシステムのモダナイゼーション
～成功の鍵は「組織」が握っている～
- AWS-04** Enterprise Systemクラウド移行最前線 SAP/MS/VMware/DB
最新アップデート
- AWS-05** リホストから始める AWS へのサーバー移行
- AWS-08** モダナイゼーションの進め方

Thank you!

畠 泰三

技術統括本部 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

