



[AWS Black Belt Online Seminar]

Amazon MQ

ソリューションアーキテクト

石井 陽介

2021/03/17

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

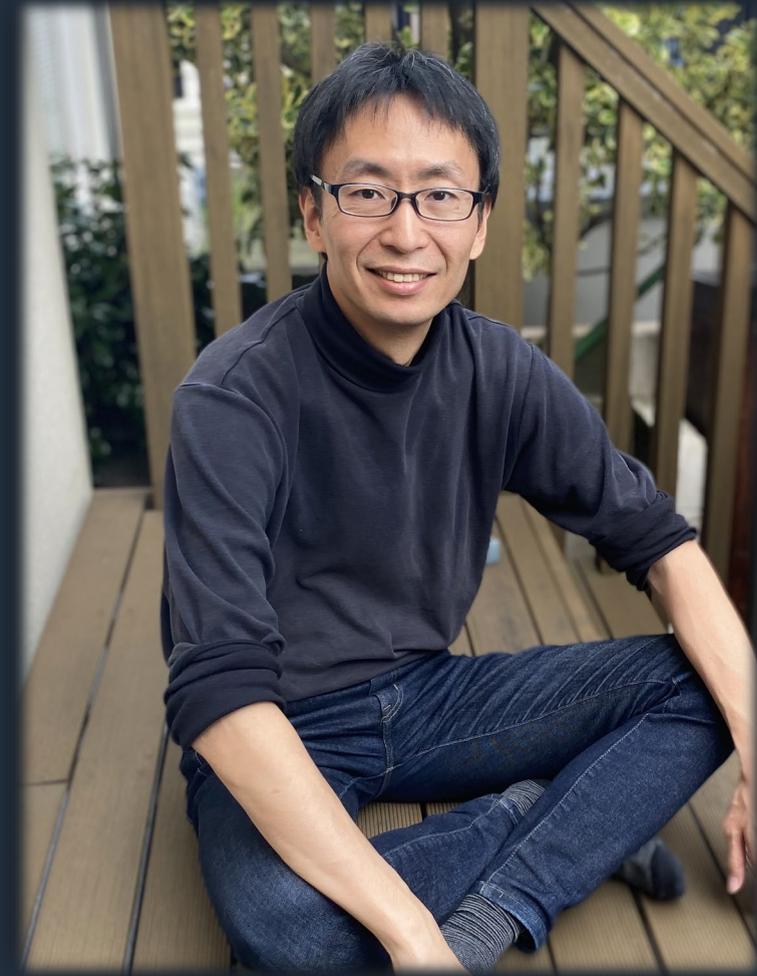
<https://amzn.to/JPArchive>



自己紹介

石井 陽介

アマゾンウェブサービスジャパン株式会社
技術統括本部 インダストリーソリューション部
ソリューションアーキテクト



📦 好きなAWSサービス



AWS Lambda



Amazon Simple Queue Service



Amazon MQ



内容についての注意点

- 本資料では2021年3月17日現在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おります。日本居住者のお客様には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.

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



ミッションクリティカルな業務を支えるメッセージング



金融サービス

取引やトランザクション



小売

受注処理やフルフィル
メント



ヘルスケア

医療データ関係



メディア・エンターテインメント

画像・動画処理



出版

文書取り込み、
出版、検索



教育

学習やエンゲージメント



製造

多段階処理

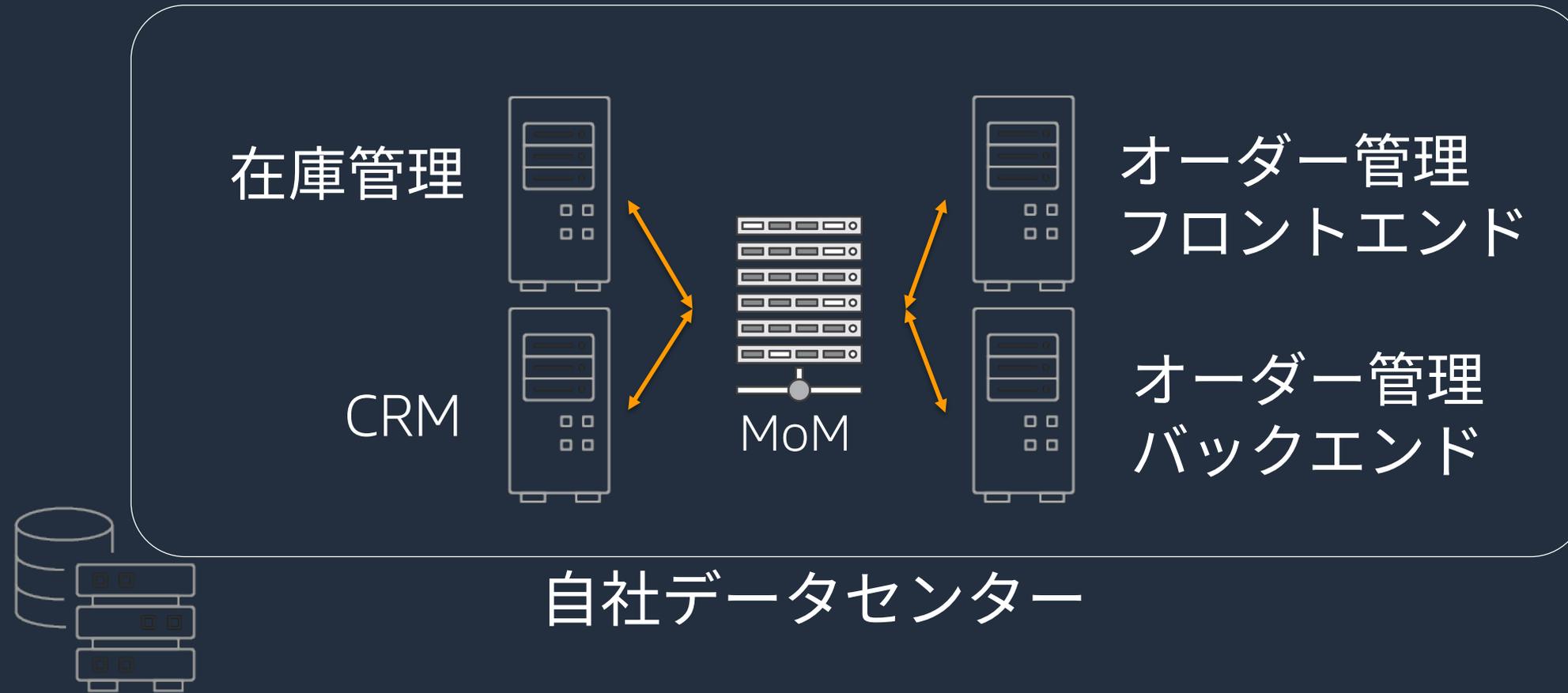


公共

市民サービス

大規模かつ高耐障害性を備えたシステム群の疎結合
相互依存性の低減 → イノベーションの加速

今日のエンタープライズメッセージング



メッセージ指向ミドルウェア (MoM) または **メッセージブローカー**

メッセージブローカー

- ソフトウェア・コンポーネント間で情報の伝達を容易に
- 各コンポーネントを疎結合化し、相互に依存性を最小化
- 投入データのバッファリングし、バルク処理を容易に
- それぞれが最も効率的に処理を行えるようスケール、バルク処理等を適用

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



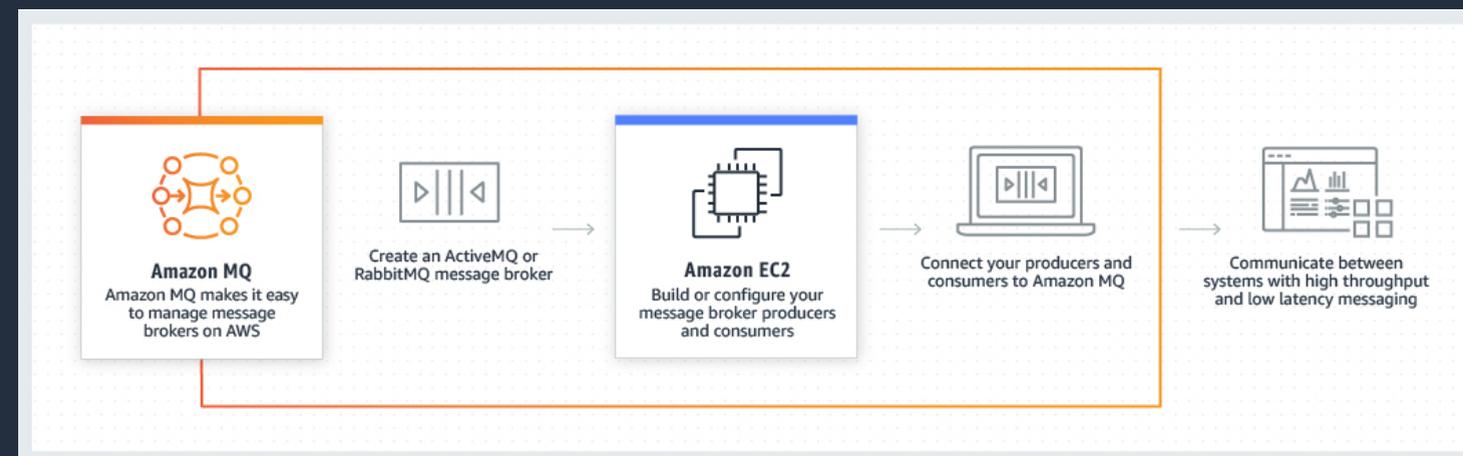
Amazon MQ

オープンソースメッセージブローカー向け完全マネージド型サービス

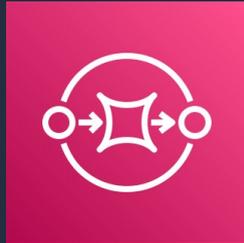
- Apache ActiveMQ, RabbitMQ をサポート
- 運用負荷の軽減
- 豊富なメッセージング機能
- 低遅延および高耐久性
- 業界標準のAPIやプロトコルに対応



Amazon MQ



AWSのメッセージングサービス



Amazon Simple
Queue Service (SQS)

マイクロサービス、分散システム、およびサーバーレスアプリケーション用の**完全マネージド型メッセージキュー**

自動的にスケール

リクエスト数に応じたコスト
100万リクエストごとに\$0.50~



Amazon Simple
Notification Service (SNS)

フルマネージド型 pub/sub メッセージング、SMS、電子メール、およびモバイルプッシュ通知

自動的にスケール

リクエスト数に応じたコスト
100万リクエストごとに\$0.50~



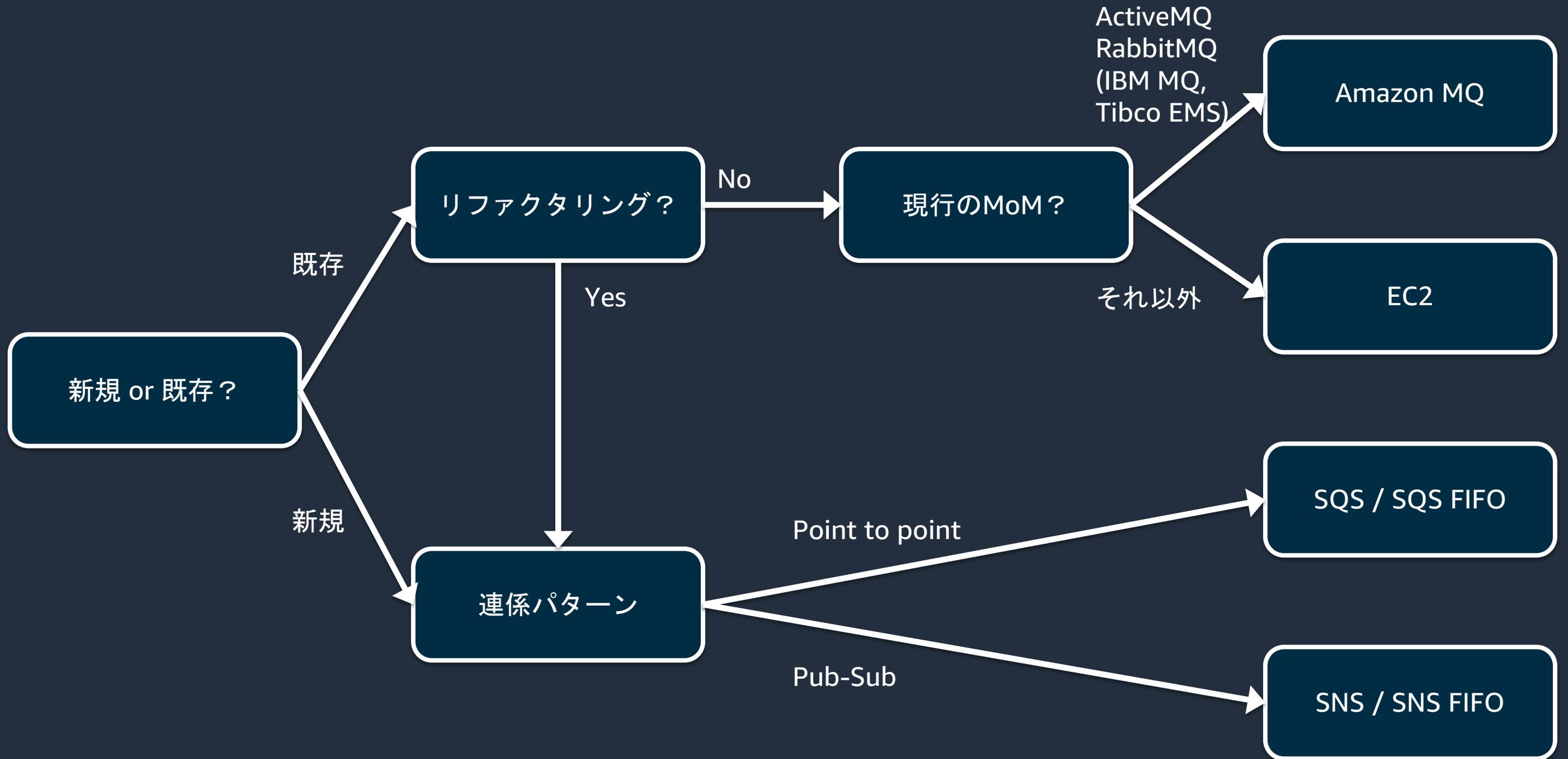
Amazon MQ

Apache ActiveMQ および RabbitMQ 向けの完全マネージド型サービス

垂直および水平スケール*
(* Network of Brokersによる)

インスタンスと
ストレージに応じたコスト
時間あたり\$0.03~
GB・月あたり\$0.12~

AWSのメッセージングサービスの使い分け



AWSのメッセージングサービスの使い分け

Amazon SQS & Amazon SNS

- クラウドネイティブなアプリケーション
- シンプル
- 無制限のスループット
- フルマネージド

Amazon MQ

- アプリケーション移行
- APIの互換性
- 豊富なメッセージング機能
- 無制限のメッセージサイズ
- 無制限のメッセージ保持期間
- 有限のスケール
- マネージドインフラストラクチャ

Amazon MQ の主なアップデート

2017

Amazon MQ GA
ActiveMQ 5.15.0
mq.t2.micro broker instance
mq.m4.large broker instance

2018

AWS CloudFormation
mq.m5 broker instances
SOC & HIPAA compliance
ActiveMQ 5.15.8
Network of Brokers

2019

ActiveMQ 5.15.9
Tag-based policies
Server-side encryption
FIPS compliance
EBS-based throughput-optimized brokers

2020

ActiveMQ 5.15.12
mq.t3.micro broker instance
LDAP integration
RabbitMQ 3.8.6

本日の内容について

本日のセッションでは Amazon MQ for Apache ActiveMQ に関する情報はアップデートと Amazon MQ for RabbitMQ との対比で重要となる点を中心にお伝えいたします。

Amazon MQ for Apache ActiveMQ の詳細は以下のリンクから過去の Blackbelt online seminar をご参照ください

- 資料
https://d1.awsstatic.com/webinars/jp/pdf/services/20190703_AWS-Blackbelt_AmazonMQ.pdf
- 動画
<https://youtu.be/-qHefQ3JoX8>

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



Apache ActiveMQ 5 概要



多様な関係パターンをサポートする オープンソースメッセージブローカー

- Javaで実装され JMS, REST, WebSocketのインターフェースを提供
- JMS1.1, J2EE 1.4 をフルサポート
永続的メッセージ、分散トランザクション (XA)、リクエスト/リプライ、セレクトター 等を利用可能
- AMQP, MQTT, OpenWire, STOMP に対応
- Java, C, C++, C#, Ruby, Perl, Python, PHP を始め、多様な言語から接続可能

Cross Language Clients: <https://activemq.apache.org/cross-language-clients>



Code Samples

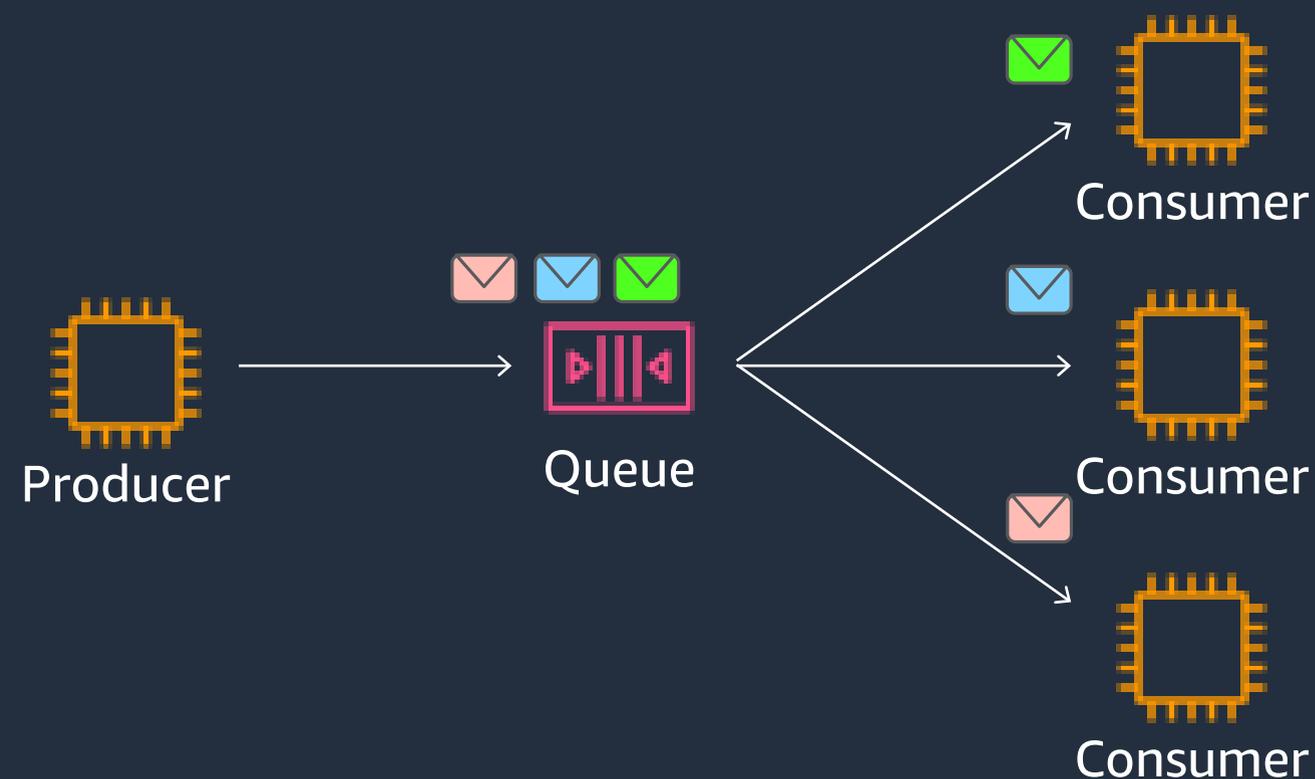
Language	Variant	Interface	Protocol	Links
Awk	Gawk	Sockets	STOMP	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
C/C++	None	Library	STOMP	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
COBOL	GNU COBOL	Sockets	STOMP	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
COBOL	Visual COBOL	JMS	OpenWire	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
Ceylon	None	JMS	OpenWire	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
Clojure	None	JMS	OpenWire	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
D	None	Sockets	STOMP	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
Dylan	Open Dylan	Sockets	STOMP	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic
Erlang	None	Library	STOMP	Send to Queue , Send to Topic , Receive from Queue , Receive from Topic

基本的な構成要素 - キュー



Point-to-Point で信頼性のあるメッセージングを実現

- プロデューサーが送信したメッセージはキューに一時的に保管される
- メッセージの処理準備が整ったコンシューマーがキューからメッセージを取得
- 各メッセージは単一のコンシューマーのみが取得
- メッセージの順序性 (FIFO: first-in-first-out) と once-and-only-once 配信をサポートする

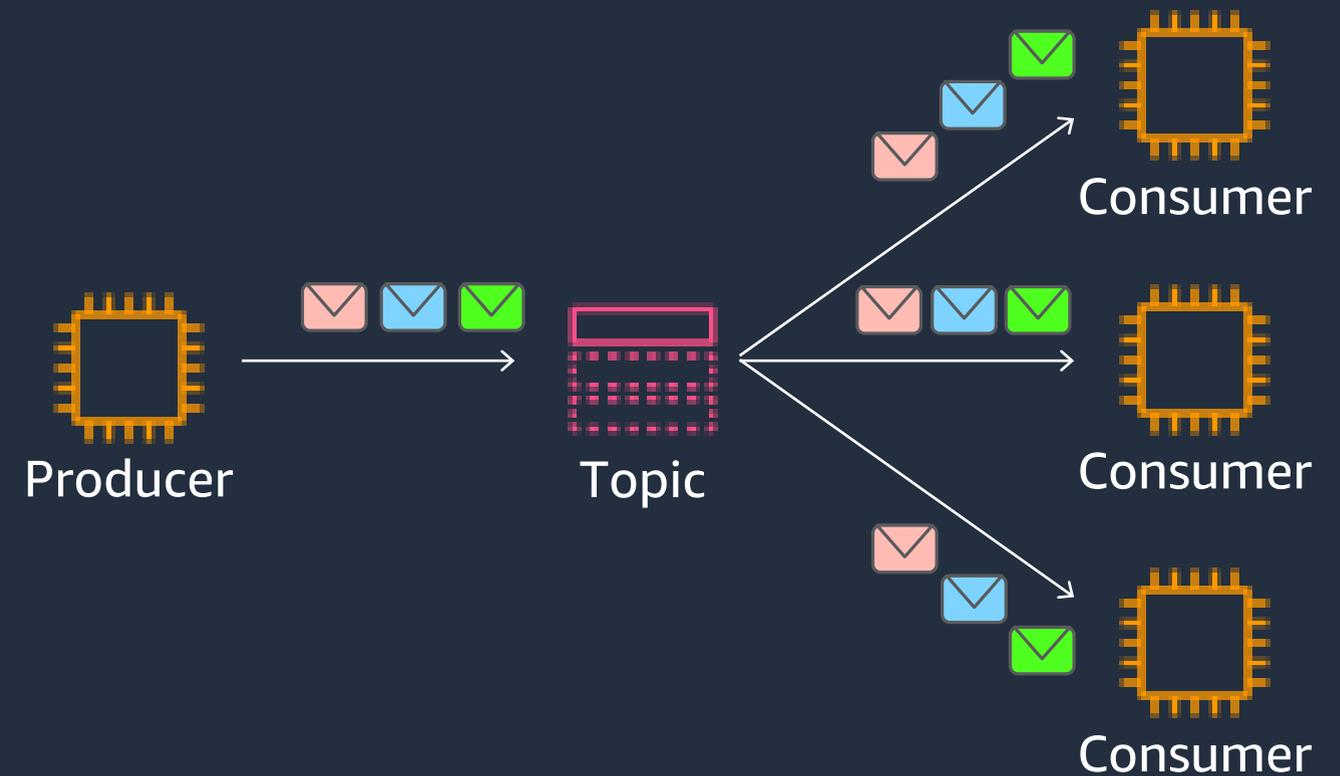


基本的な構成要素 – トピック



同一メッセージを複数の送信先へ送達

- パブリッシュ/サブスクライブ型
- メッセージはアクティブなサブスクリプションを持つ全てのコンシューマーに配信される
- 永続サブスクリプション (Durable Subscription) を使用すると切断中のメッセージを再接続時に取得可能



基本的な構成要素 – ブローカー



Amazon MQ for Apache ActiveMQ で実行されるメッセージブローカー環境

- 複数バージョンのブローカーエンジン
- 複数のインスタンスタイプ
- 単一ブローカー または マルチ AZ アクティブ/スタンバイブローカー
- パブリック接続可能 または パブリック接続不可
- セキュリティグループやユーザー、認可マップによるアクセス制御



ブローカー – エンジンバージョンとプロトコル



サポートするBrokerバージョン

- ActiveMQ 5.15.13
- ActiveMQ 5.15.12
- ActiveMQ 5.15.10
- ActiveMQ 5.15.9
- ActiveMQ 5.15.8
- ActiveMQ 5.15.6
- ActiveMQ 5.15.0

サポートするプロトコル

- AMQP
- MQTT
- MQTT over WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

ブローカー - インスタンスタイプ



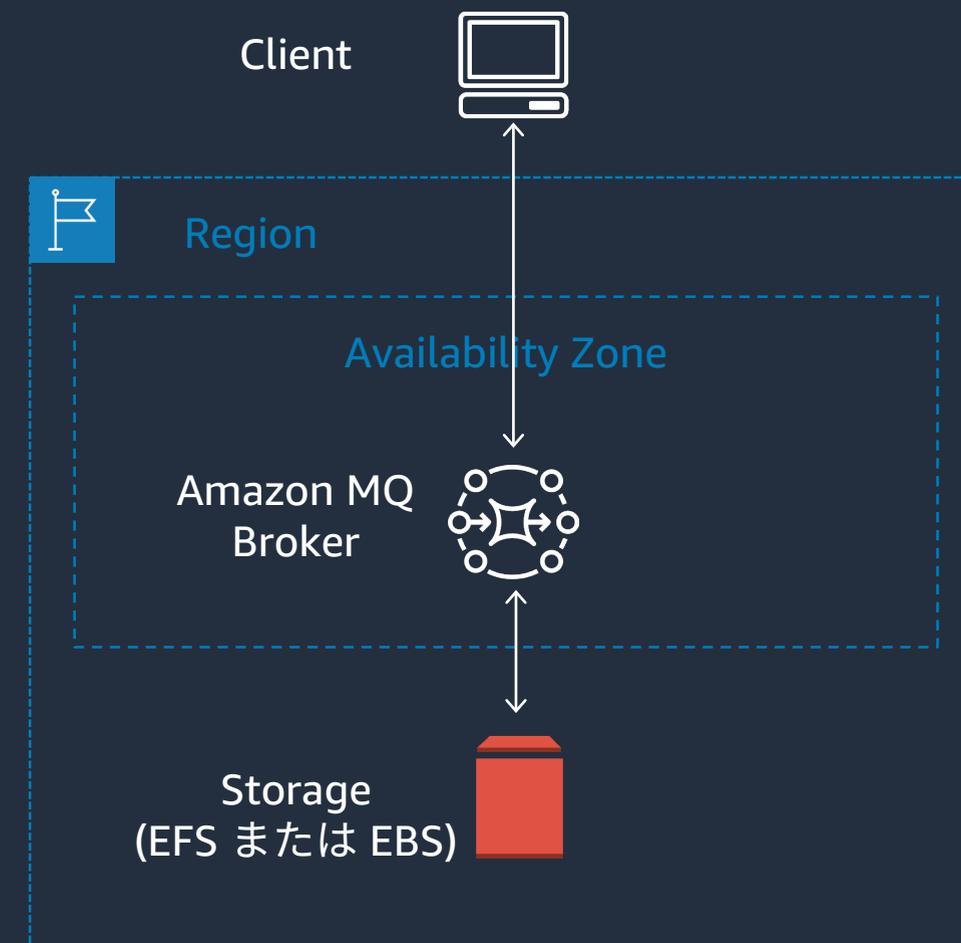
インスタンスタイプ	vCPU	メモリ (GiB)	ネットワークパフォーマンス	コメント
mq.t3.micro	2	1	低	基本的な評価用途に適する。単一インスタンスブローカーの場合は AWS 無料利用枠の対象 。
mq.t2.micro	1	1	低	
mq.m5.large	2	8	高	一般的な開発、テスト、および本稼働のワークロードの用途に適する。
mq.m5.xlarge	4	16	高	高いスループットを必要とする一般的な開発、テスト、および本稼働のワークロードに適する。
mq.m5.2xlarge	8	32	高	
mq.m5.4xlarge	16	64	高	
mq.m4.large	2	8	中	既存のブローカーのデプロイとの互換性が必要な場合に使用。新規には mq.m5.* を推奨。

単一インスタンスブローカー



1 アベイラビリティゾーン、1ブローカーのシンプルな構成

- 開発やテスト用に向く
- 無償利用枠の対象
- SLA (Service Level Agreement) は定義されない
- mq.m5 インスタンスファミリーの場合、ストレージを選択可能
 - **EFS**: AZ間でデータをレプリケートすることで耐久性を実現
 - **EBS**: スループットに最適化されており、AZ内の複数サーバーにデータをレプリケート

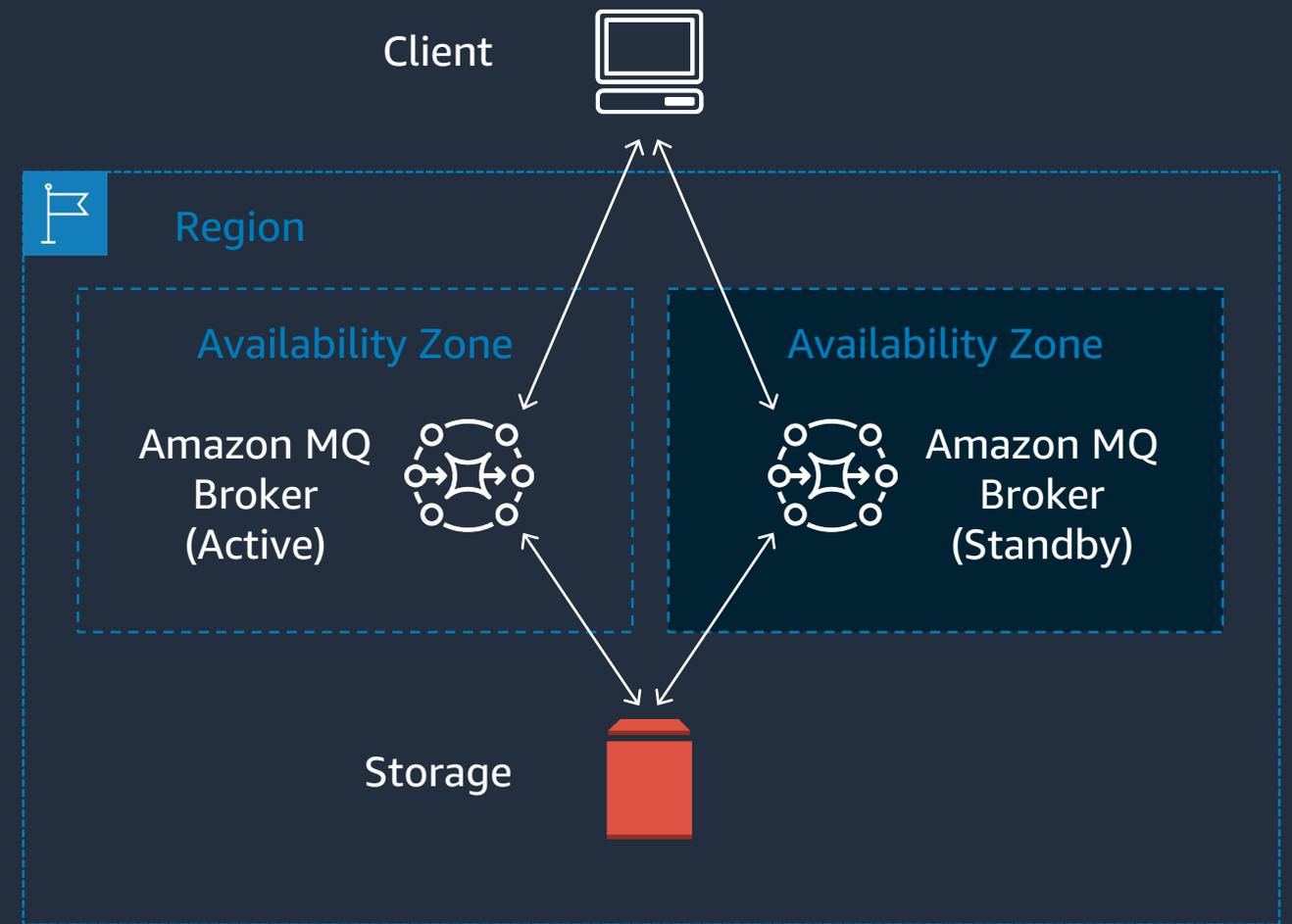


アクティブ/スタンバイローカー



2つのアベイラビリティゾーンに分散配置される高可用性対応構成

- 各プロトコルや ActiveMQ ウェブコンソールのエンドポイントはアクティブインスタンス側のみが利用可能
- 障害発生時にはスタンバイ側へテイクオーバー
- アプリケーションは**フェイルオーバートランスポート**を定義することで稼働中のエンドポイントへ接続
- SLA (Service Level Agreement) の対象



フェイルオーバートランスポート



複数エンドポイントを指定し、障害時に接続をフェイルオーバー

- クライアントでの接続先指定時に複数のエンドポイントを列挙可能
- デフォルトではランダムに接続先が選択される
- 接続に失敗した場合、リストから他のエンドポイントをランダムに選択

フォーマット

```
failover:(<uri1>, ..., <uriN>)?<transportOptions>&<nestedURIOptions>
```

<transportOptions>: フェイルオーバートランスポートに適用するオプション

<nestedURIOptions>: トランスポート内の全URIに適用するオプション (nested.* 形式で記載)

```
static final String DESTINATION_STRING = "failover:(ssl://***-1.mq.us-west-2.amazonaws.com:61617,ssl://***-2.mq.us-west-2.amazonaws.com:61617)?timeout=3000&nested.wireFormat.maxInactivityDuration=1000";  
ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory(DESTINATION_STRING);
```

java

The Failover Transport: <https://activemq.apache.org/failover-transport-reference>

基本的な構成要素 – ブローカーのネットワーク



複数のブローカーインスタンスを組み合わせ
て高度なメッセージング要件を実現

- 複数のブローカーで構成
- ネットワークコネクタを使用してブローカー同士を接続することによって確立
- ブローカーインスタンスを追加することで、総スループットと最大クライアント接続数の拡張が可能
- クライアントが複数のアクティブなブローカーインスタンスを認識できるようにすることで、可用性を向上させる
- メッセージング要件に応じて、さまざまなトポロジ（コンセントレータ、ハブアンドスポーク、メッシュなど）で設定可能

ブローカーのネットワーク - メッシュトポロジ



オンデマンドにすべてのブローカー間でメッセージを転送

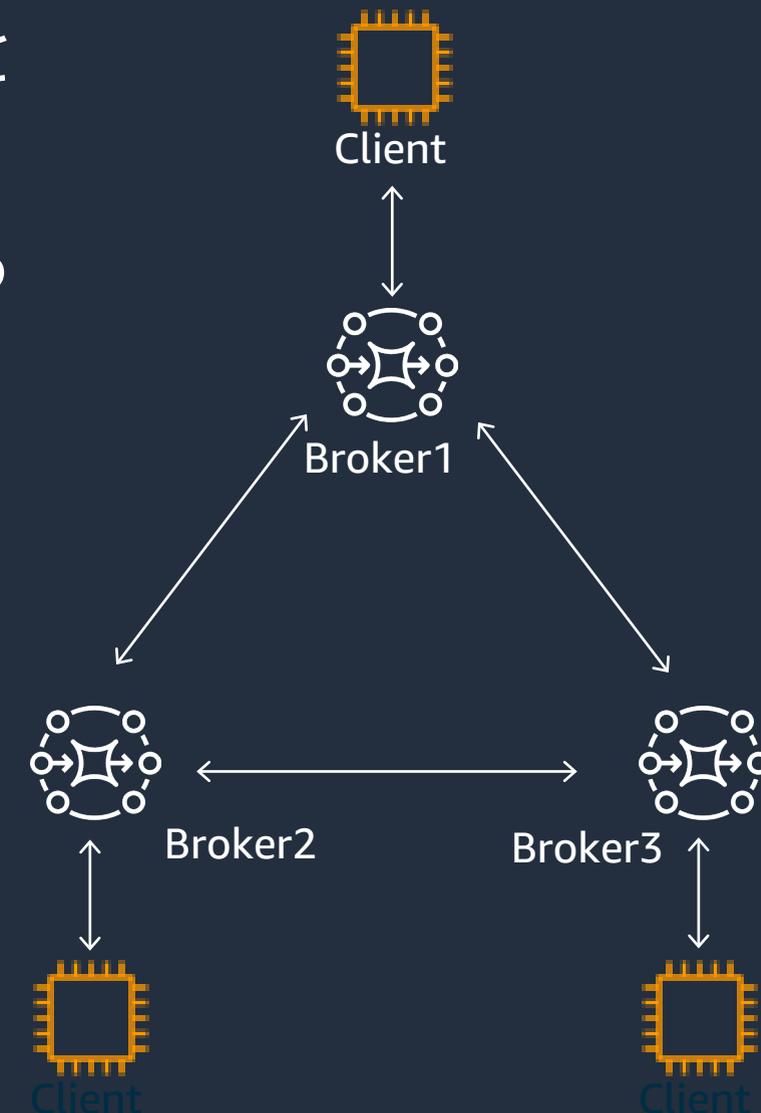
- 接続されたブローカー間で相互にメッセージを転送
- 地理的に分散した拠点間の通信等で使用される

```
<networkConnectors>
  <networkConnector name="con1to2" userName="user" duplex="true"
    uri="static:(ssl://broker2-hostname:61617)" />
  <networkConnector name="con1to3" userName="user" duplex="true"
    uri="static:(ssl://broker3-hostname:61617)" />
</networkConnectors>
```

activemq.xml (Broker1)

```
<networkConnectors>
  <networkConnector name="con2to3" userName="user" duplex="true"
    uri="static:(ssl://broker3-hostname:61617)" />
</networkConnectors>
```

activemq.xml (Broker2)



基本的な構成要素 – 設定



Amazon MQ ブローカーの挙動を定義する XML ファイル

- ActiveMQ の activemq.xml と類似
- エンジン毎に XML Schema が提供される
- AWS マネジメントコンソールから直接編集、またはCLIからアップロード
- 更新時にリビジョンが作成され、変更履歴の追跡やロールバックが可能

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<broker
xmlns="http://activemq.apache.org/schema/core">
    <destinationInterceptors ... />
    <persistenceAdapter ... />
    <destinationPolicy ... />
    <destinations ... />
    <plugins ... />
    <networkConnectors ... />
</broker>
```

Amazon MQ ブローカー設定パラメータ

https://docs.aws.amazon.com/ja_jp/amazon-mq/latest/developer-guide/amazon-mq-broker-configuration-parameters.html

Amazon MQ で許容される設定オプション例 (抜粋)



plugins

- authorizationPlugin
- discardingDLQBrokerPlugin
- forcePersistencyModeBrokerPlugin
- redeliveryPlugin
- statisticsBrokerPlugin
- timeStampingBrokerPlugin

destinationInterceptors

- mirroredQueue
- virtualTopic
- compositeQueue

destinationPolicy

- pendingMessageLimitStrategy
- dispatchPolicy
- subscriptionRecoveryPolicy

Amazon MQ ブローカー設定パラメータ

https://docs.aws.amazon.com/ja_jp/amazon-mq/latest/developer-guide/amazon-mq-broker-configuration-parameters.html

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- **Amazon MQ for RabbitMQ**
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



RabbitMQ 概要



企業規模を問わず支持を集める軽量なメッセージブローカー

- Erlangで実装された軽量ブローカー
- プロトコルとして AMQP に標準対応
- プラグインによる拡張性
- Java, .NET, Ruby, Python, PHP, JavaScript を始め、多様な言語から接続可能



Clients Libraries and Developer Tools

Overview

RabbitMQ is [officially supported](#) on a number of operating systems and has several official client libraries. In addition, the RabbitMQ community has created numerous clients, adaptors and tools that we list here for your convenience.

Please [contact us](#) with suggestions for things you would like to see added to this list.

Java and Spring

Java

- [RabbitMQ Java client](#) and its [developer guide](#)
- [RabbitMQ JMS client](#)
- [Reactor RabbitMQ](#), a reactive API for RabbitMQ based on [Reactor](#) and [RabbitMQ Java Client](#)
- [camel-rabbitmq](#), an Apache Camel component for interacting with RabbitMQ. This ships as part of Camel 2.12 or later.
- [Banyan](#), a RabbitMQ-backed message bus with a tree topology.

Spring Framework

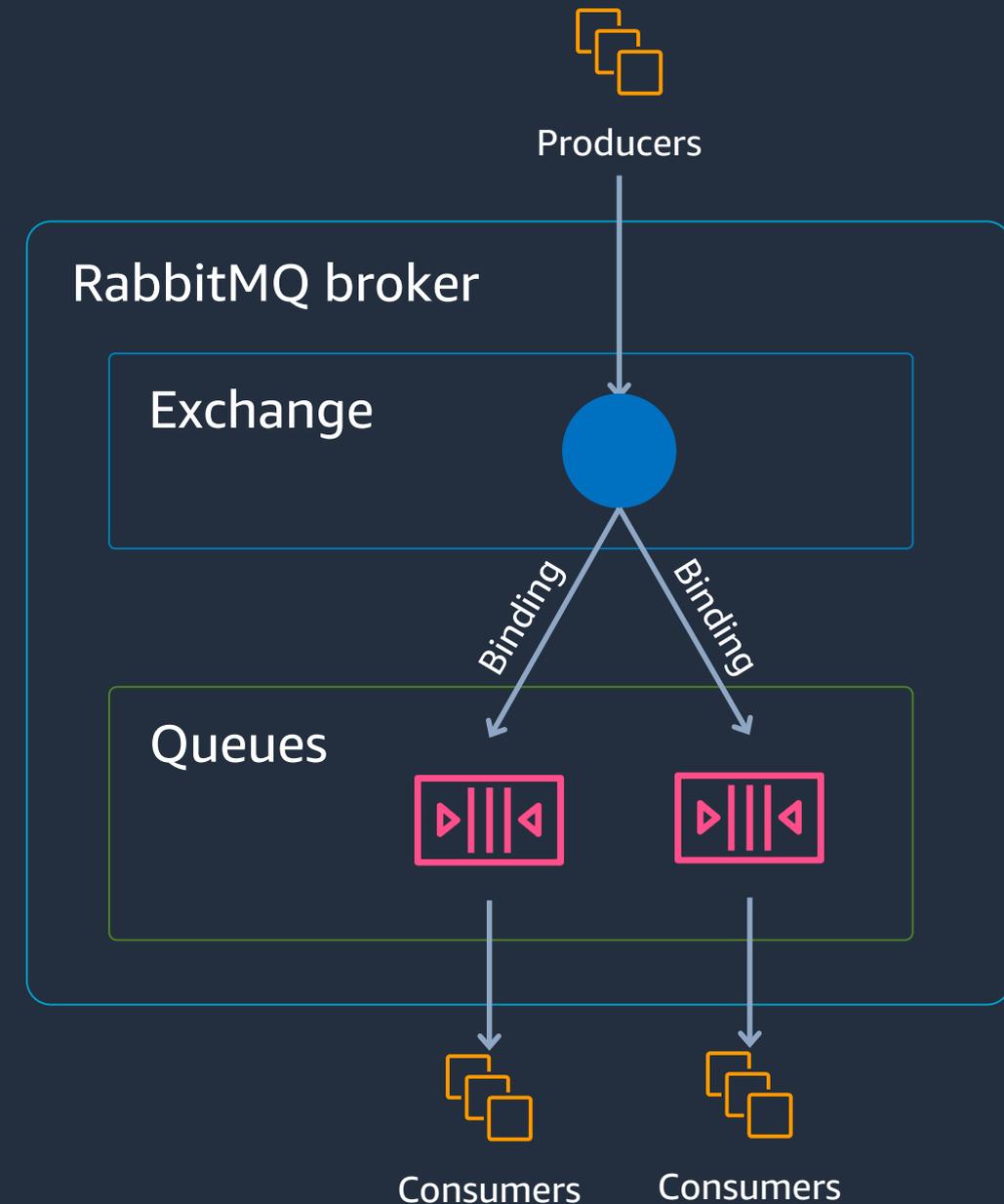
- [Spring AMQP project for Java](#)
- [Spring Cloud Data Flow](#)
- [Spring Integration](#)

Clients Libraries and Developer Tools: <https://www.rabbitmq.com/devtools.html>

RabbitMQ の主なメッセージング要素



- **Exchanges** はメッセージを 0 以上の Queue へルーティングします
Exchange Types:
Direct, Topic, Fanout, Headers
- **Bindings** メッセージを Queue へルーティングするために Exchange によって利用されます
- **Queues** はアプリケーションが利用するメッセージを保持します



基本的な構成要素 – ブローカー



Amazon MQ for RabbitMQ で実行されるメッセージブローカー環境

- 複数のインスタンスタイプ
- 単一ブローカー または クラスター ブローカー
- パブリック接続可能 または プライベート (パブリック接続不可)



Client



Amazon MQ
ブローカー

ブローカー バージョンとプロトコル



サポートするRabbitMQバージョン

- v3.8.6

サポートするプロトコル

- AMQP (0-9-1)

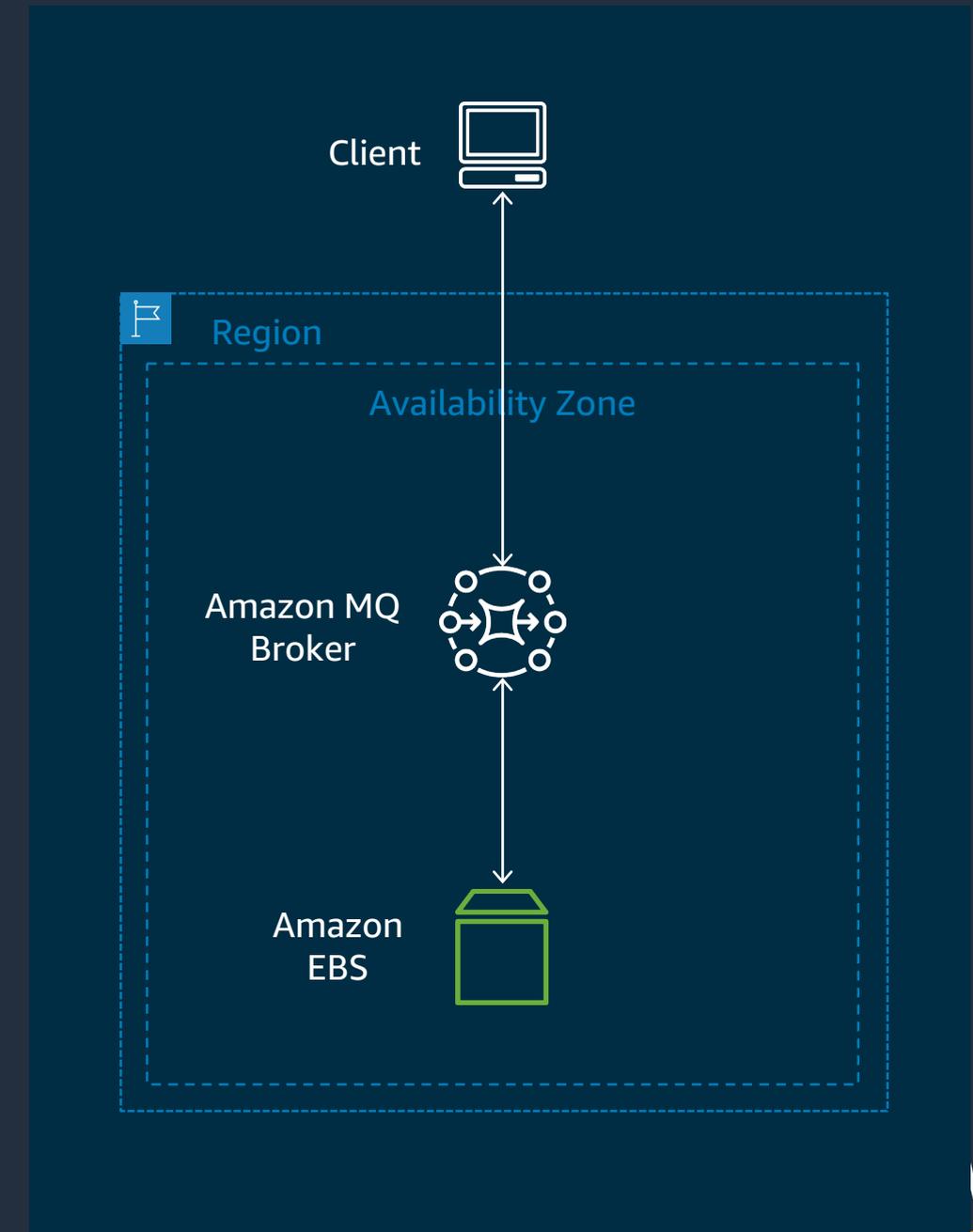
Amazon MQ の構成要素 - ブローカー/インスタンスタイプ

インスタンスタイプ	vCPU	メモリ (GiB)	ネットワークパフォーマンス	コメント
mq.t3.micro	2	1	低	基本的な評価用途に適する。クラスターのデプロイは不可。 AWS 無料利用枠 の対象。
mq.m5.large	2	8	高	一般的な開発、テスト、および本稼働のワークロードの用途に適する。
mq.m5.xlarge	4	16	高	
mq.m5.2xlarge	8	32	高	高いスループットを必要とする一般的な開発、テスト、および本稼働のワークロードに適する。
mq.m5.4xlarge	16	64	高	

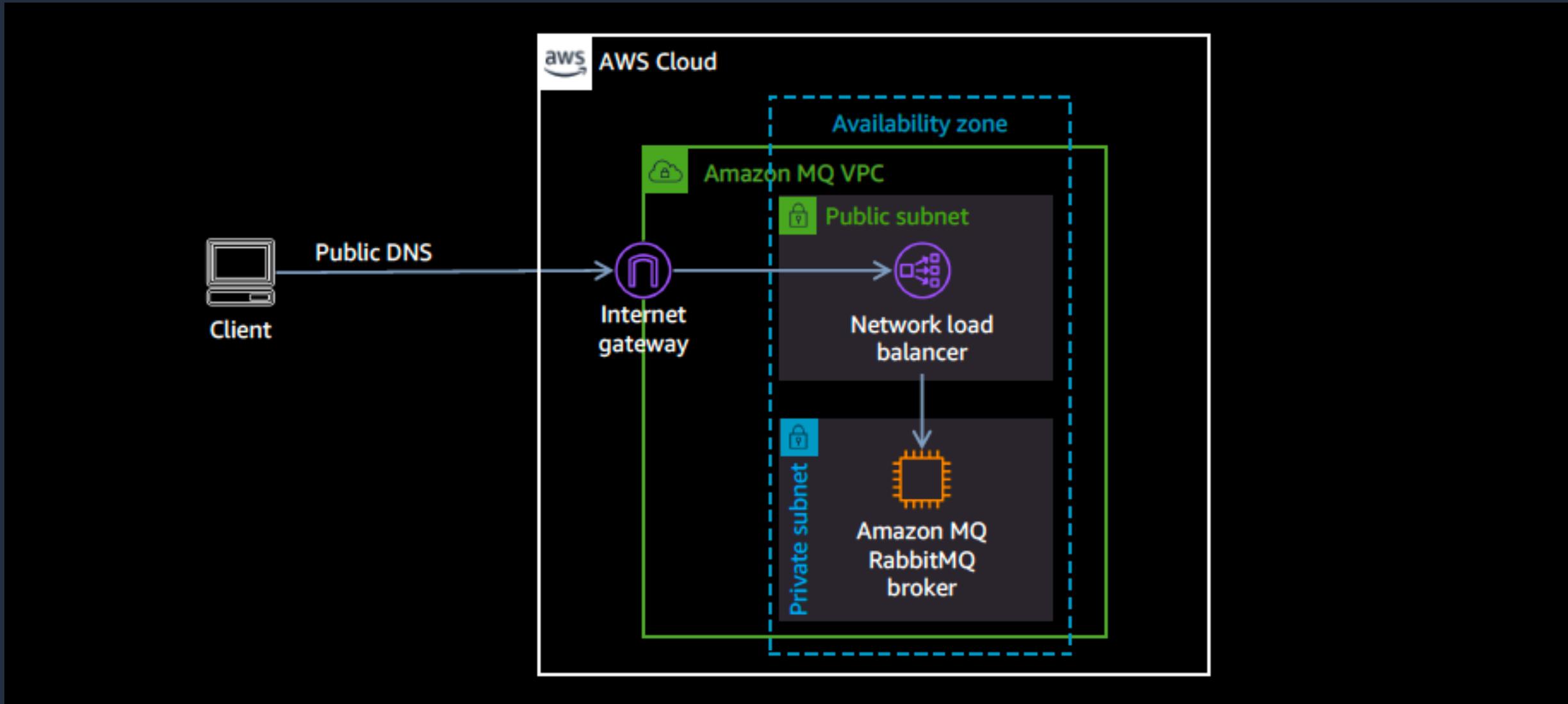
単一インスタンスブローカー



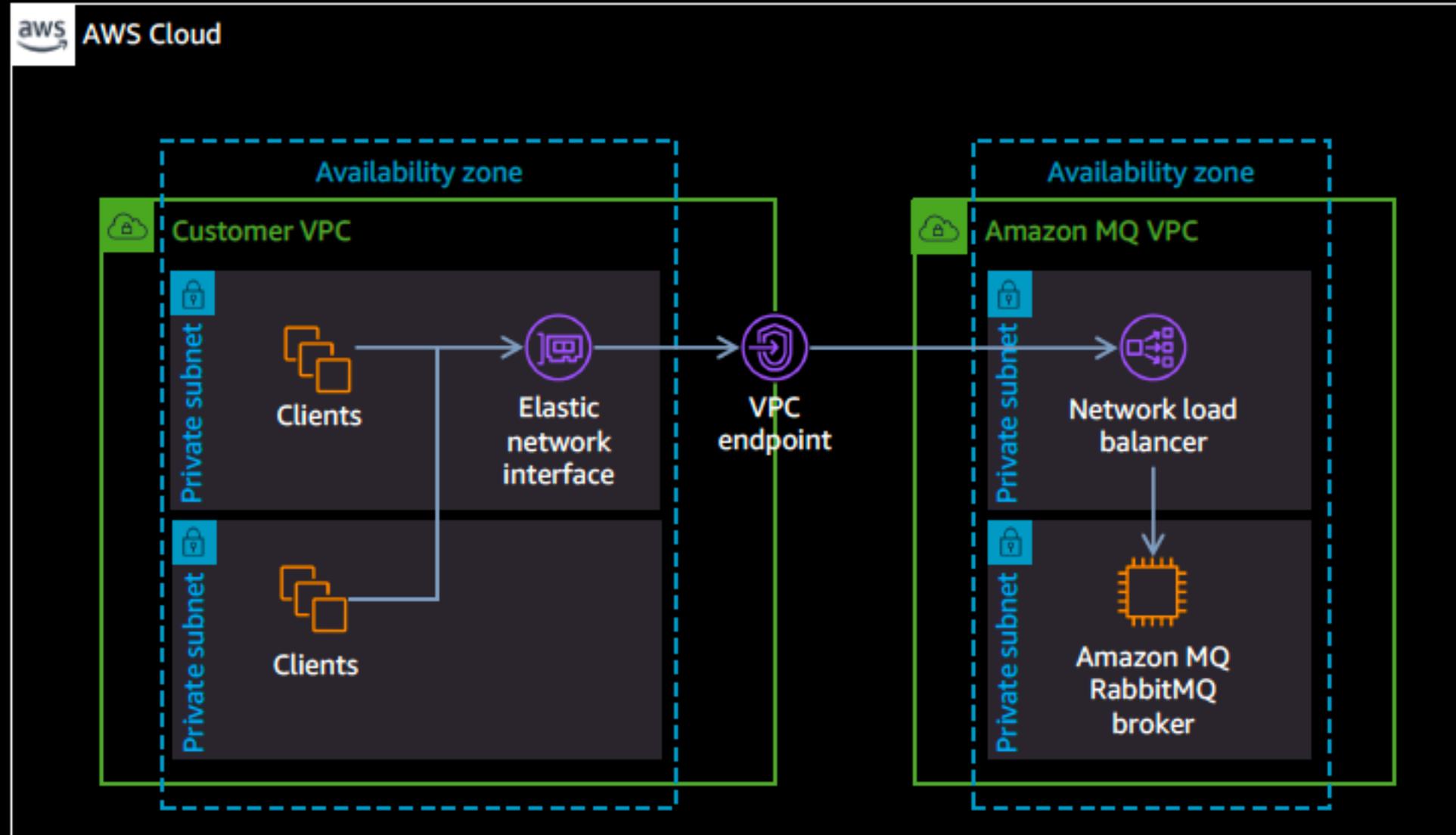
- 1つのアベイラビリティゾーンの中の1つのブローカーで構成
- NonProduction構成
- SLA (Service Level Agreement) は定義されない
- 低遅延でメッセージの永続化を必要としないユースケース向け



接続性：単一ブローカー (パブリックアクセス)



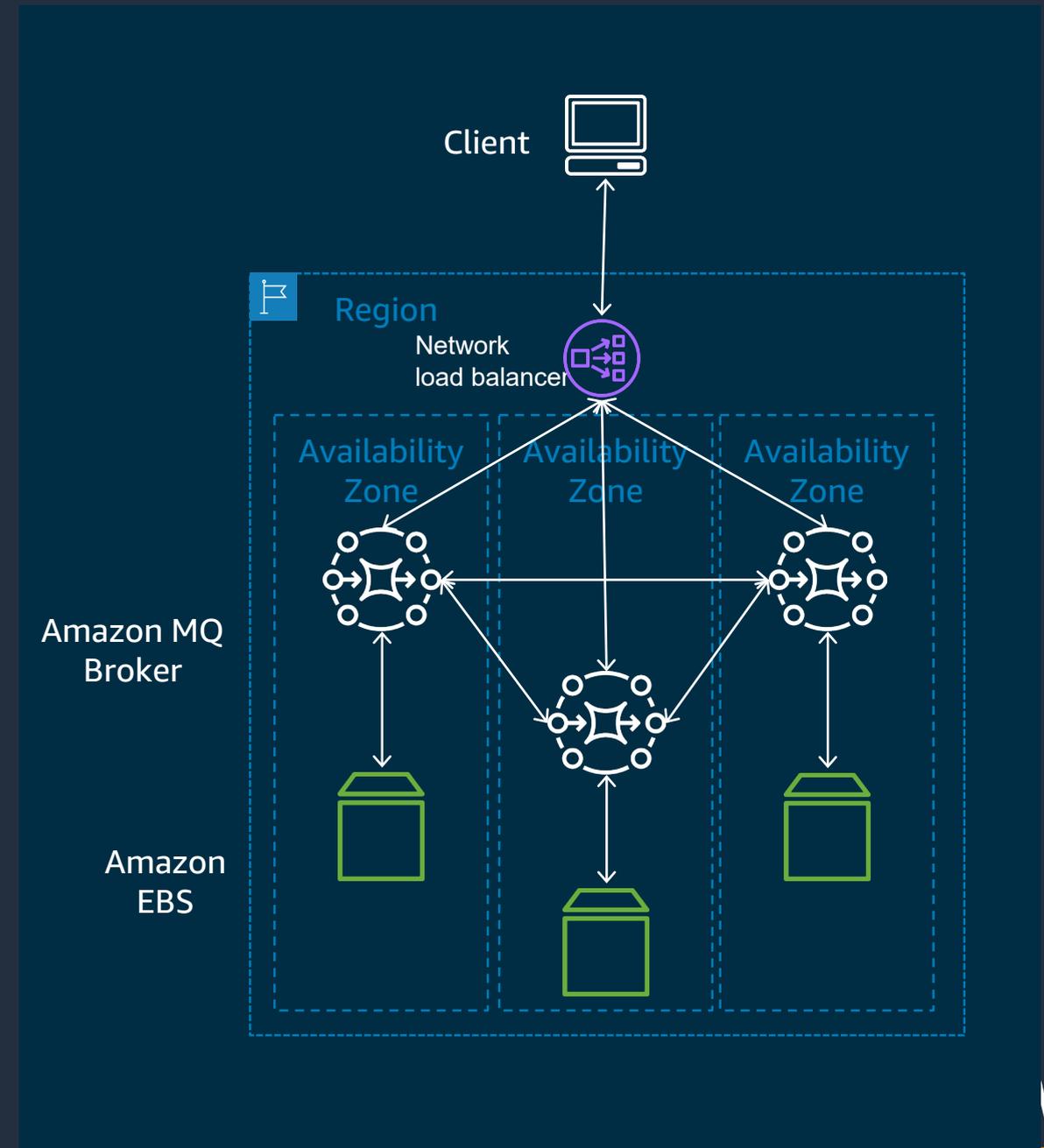
接続性：単一ブローカー (プライベートアクセス)



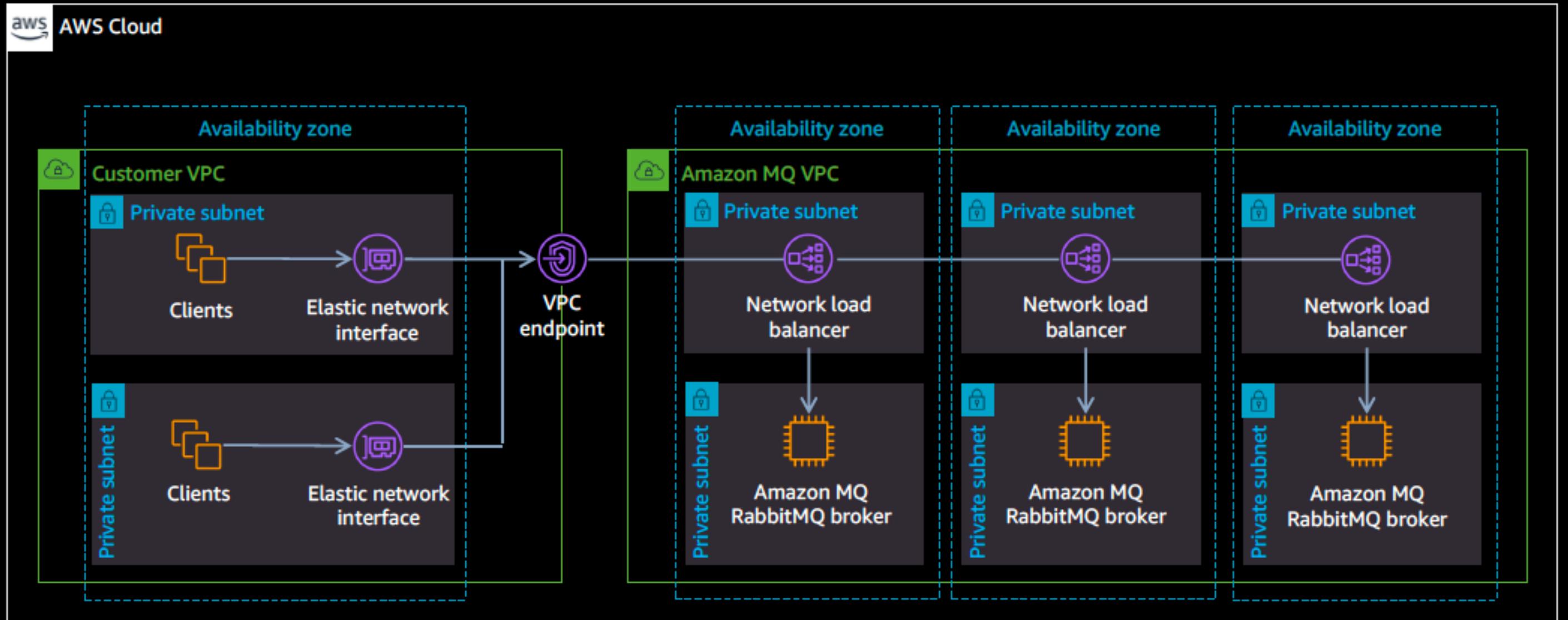
クラスターブローカー



- 3つのノードを異なるアベイラビリティゾーンに配置
- ノード間に Classic Mirrored Queue を構成
- Productionユースケース向け
- SLA (Service Level Agreement) の対象



接続性：クラスターブローカー (プライベートアクセス)



クラスターブローカーの耐久性



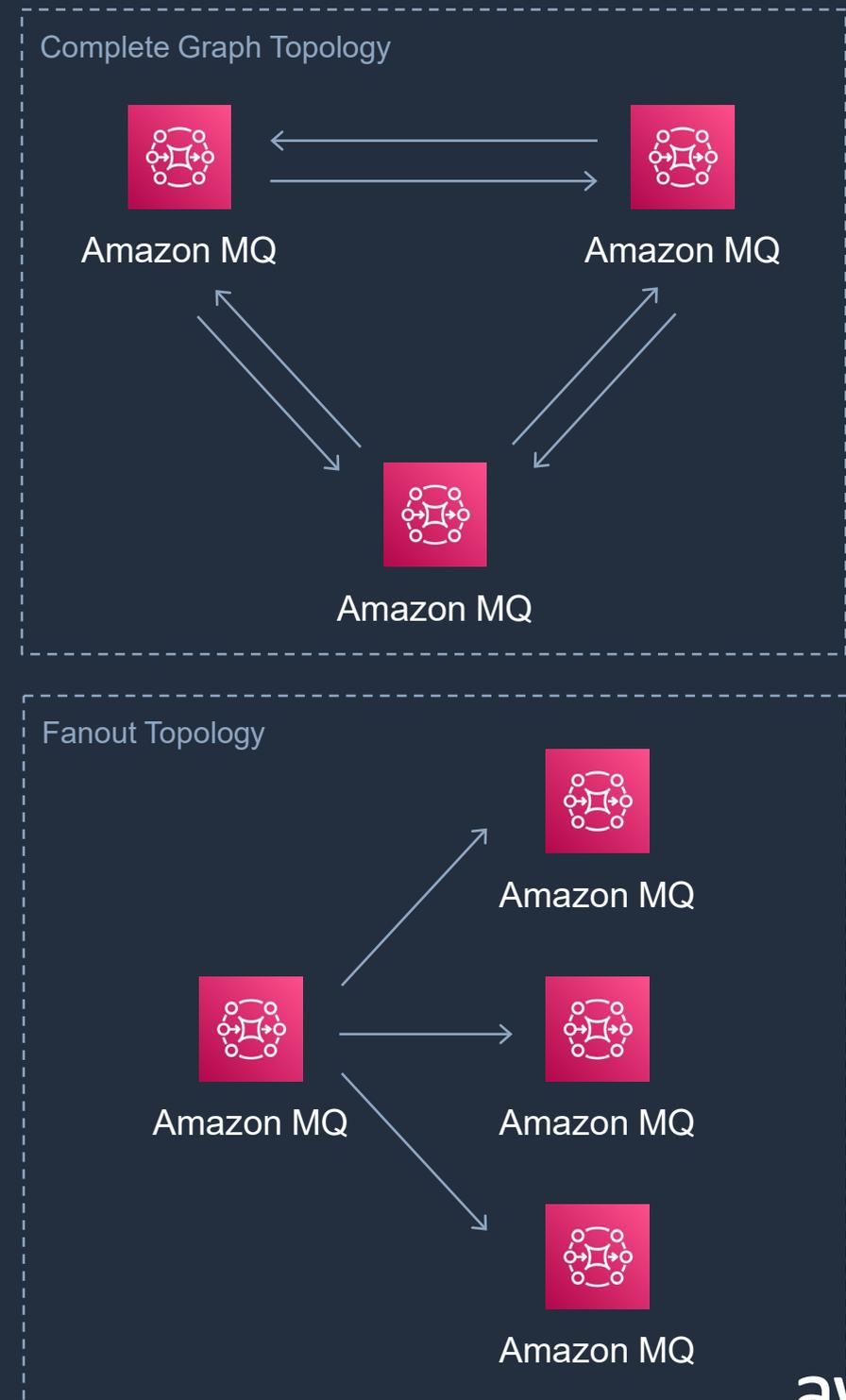
- プロデューサーおよびコンシューマー向けの単一エンドポイントを提供
- Amazon MQ が `ha-mode=all` および `ha-sync-mode=automatic` を全てのRabbitMQ ポリシーに設定することで mirrored queue 構成を実現する
- ノードの交換時には
 - Amazon MQは自動的にノードをリプレイスする
 - コンシューマーは切断されるため、再接続が必要
 - 新たな Mirror が接続されるとキューは自動的に同期される

同期中は該当キューがブロックされるのため、キューを浅く維持することが推奨

Federation Plugin



- 下流のブローカーが上流のブローカーに対して point-to-point で接続を行う
- **Federated queues** では下流のコンシューマー間で負荷を分散するために、上流のキューから下流のコンシューマーへメッセージを **移動** します
- **Federated exchanges** では pairing, complete graphs, fanout などの接続トポロジーを利用して上流の exchange から下流の exchange へメッセージを **コピー** します. **max-hops** 値によって Loop からの保護が提供されます
- 下流のブローカーが上流のブローカーに対して接続を開始するため、上流の Amazon MQ ブローカーはパブリックである必要があります



Shovel Plugin

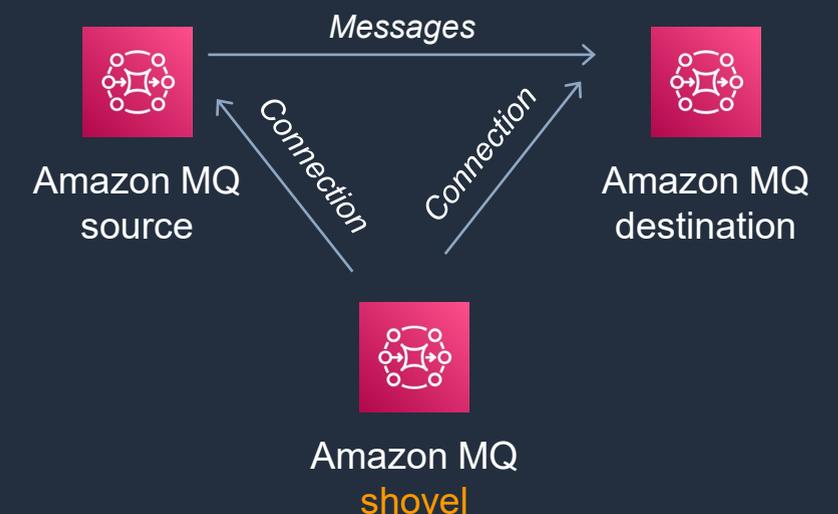


- Shovel はあたかも良くできたクライアントアプリケーションかのように、source ブローカーの queue からメッセージを取得して destination ブローカーの queue や exchange に移動します
- Shovel の接続は source ブローカー、destination ブローカー、または異なるノードやクラスターからも開始することができます
- 接続が確立された後には、Shovel の定義に従って queue、exchange、binding の宣言を自動化することもできます
- 柔軟なトポロジーが可能だが、Loop 検出は提供されない

Source broker establishes Shovel connection



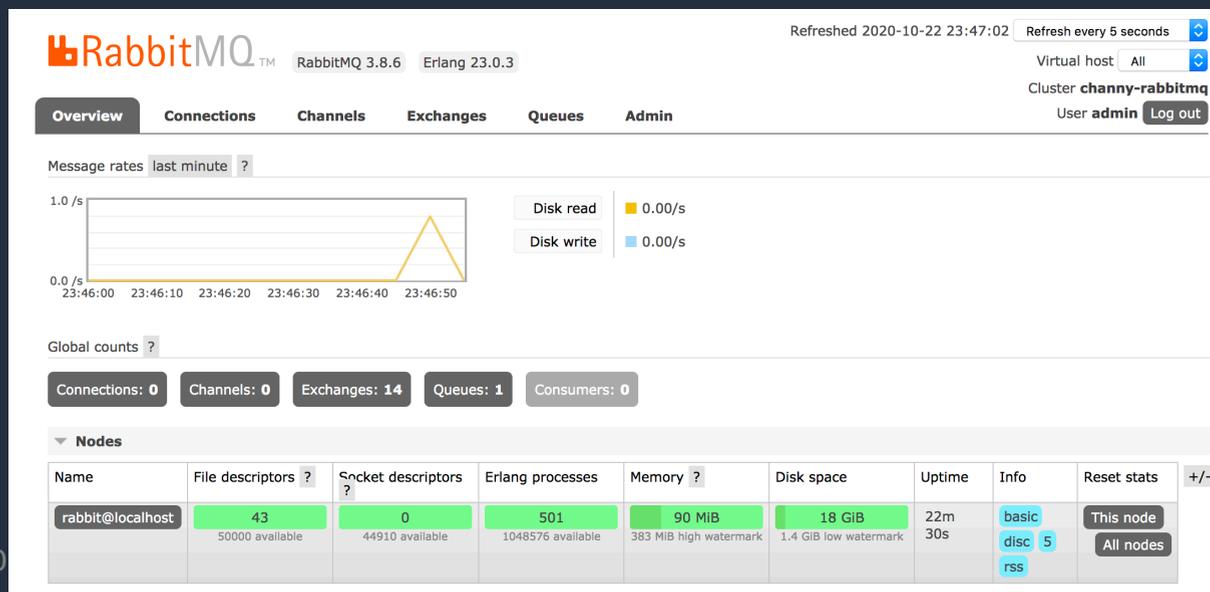
Separate broker establishes Shovel connection



Amazon MQ for RabbitMQ の管理



- RabbitMQ Web コンソール
 - グラフィカルな管理UI
- rabbitmqadmin
 - コマンドラインから利用可能な RabbitMQ REST API client
 - ブローカー作成後 https://BROKER_HOST/cli/rabbitmqadmin より RabbitMQ のバージョンに即したものを取得
 - **-P 443** Amazon MQ RabbitMQ Web コンソールのポートを指定



```
% rabbitmqadmin -s -P 443 -H BROKER_HOST ¥  
-u USERNAME -p PASSWORD list queues
```

```
+-----+-----+  
| name   | messages |  
+-----+-----+  
| my-queue | 0       |  
+-----+-----+
```

Amazon MQ for RabbitMQ の注意点



- ユーザー独自のPluginを導入することはできません
 - Management、Monitoring、Shovel、Federation の各プラグインが導入済み
- rabbitmq.conf や advanced.config に相当する設定オブジェクトは存在しません

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他





ブローカーのメンテナンスに関するポリシーを指定

- 自動マイナーバージョンアップの可否
- メンテナンスウィンドウの選択
 - 主に設定変更の適用や、ブローカーへのパッチ適用が行われる
 - 毎週2時間の枠を指定（毎回メンテナンスが行われるとは限らない）

メンテナンス

自動マイナーバージョンアップグレードの有効化
リリース時に新しいバージョンへの自動アップグレードを有効にします。自動アップグレードは、ブローカーのメンテナンスウィンドウ時に行われます。

メンテナンスウィンドウ
Amazon MQ で 保留中の変更を適用するか、ブローカーにパッチを適用する期間を選択します。

指定なし
Amazon MQ によってメンテナンスウィンドウが選択されます。

メンテナンスウィンドウの選択

開始日	開始時刻	所要時間
日曜日 ▼	00 ▼ : 00 ▼ UTC	2 時間



AWS サービスと関係したモニタリングやオペレーションが可能

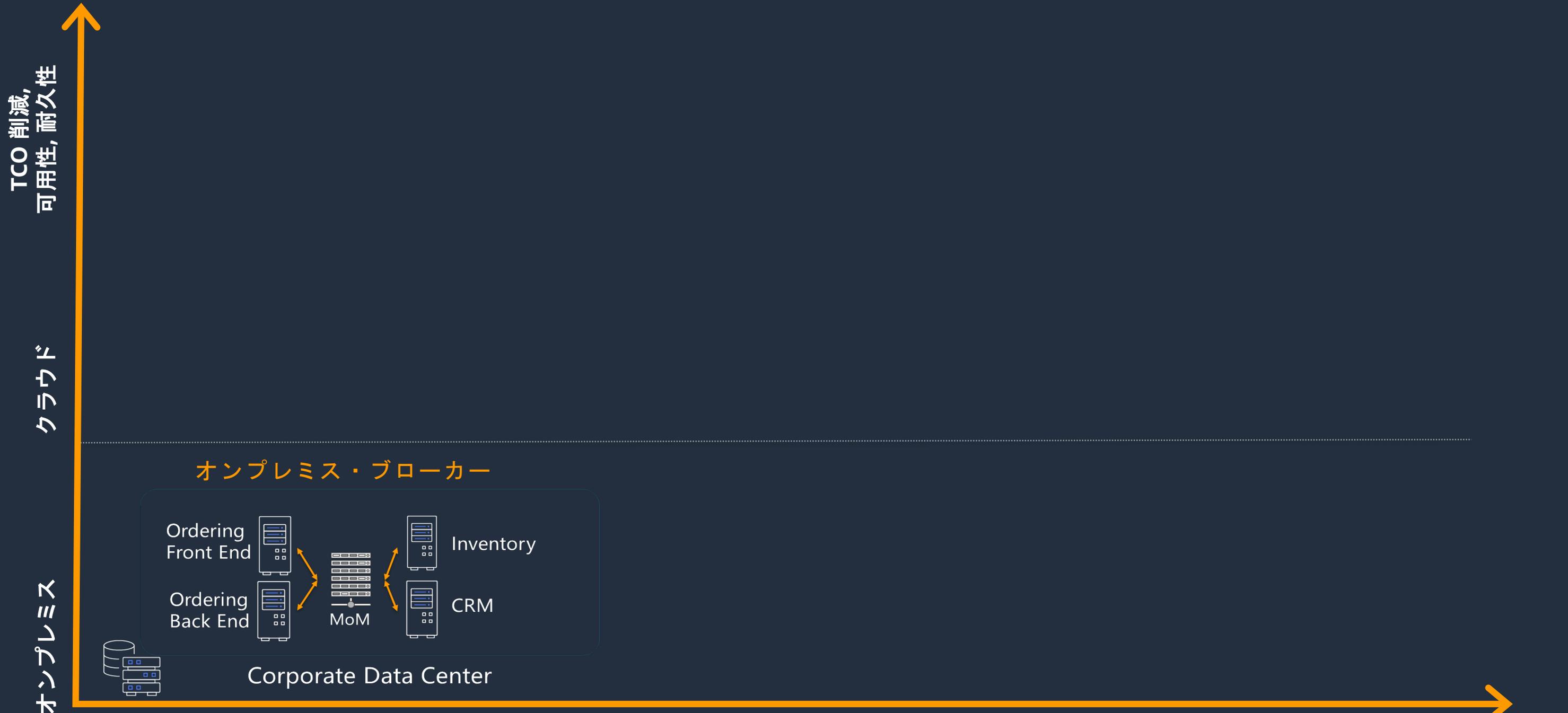


本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



アプリケーションの移行パス: オンプレミス

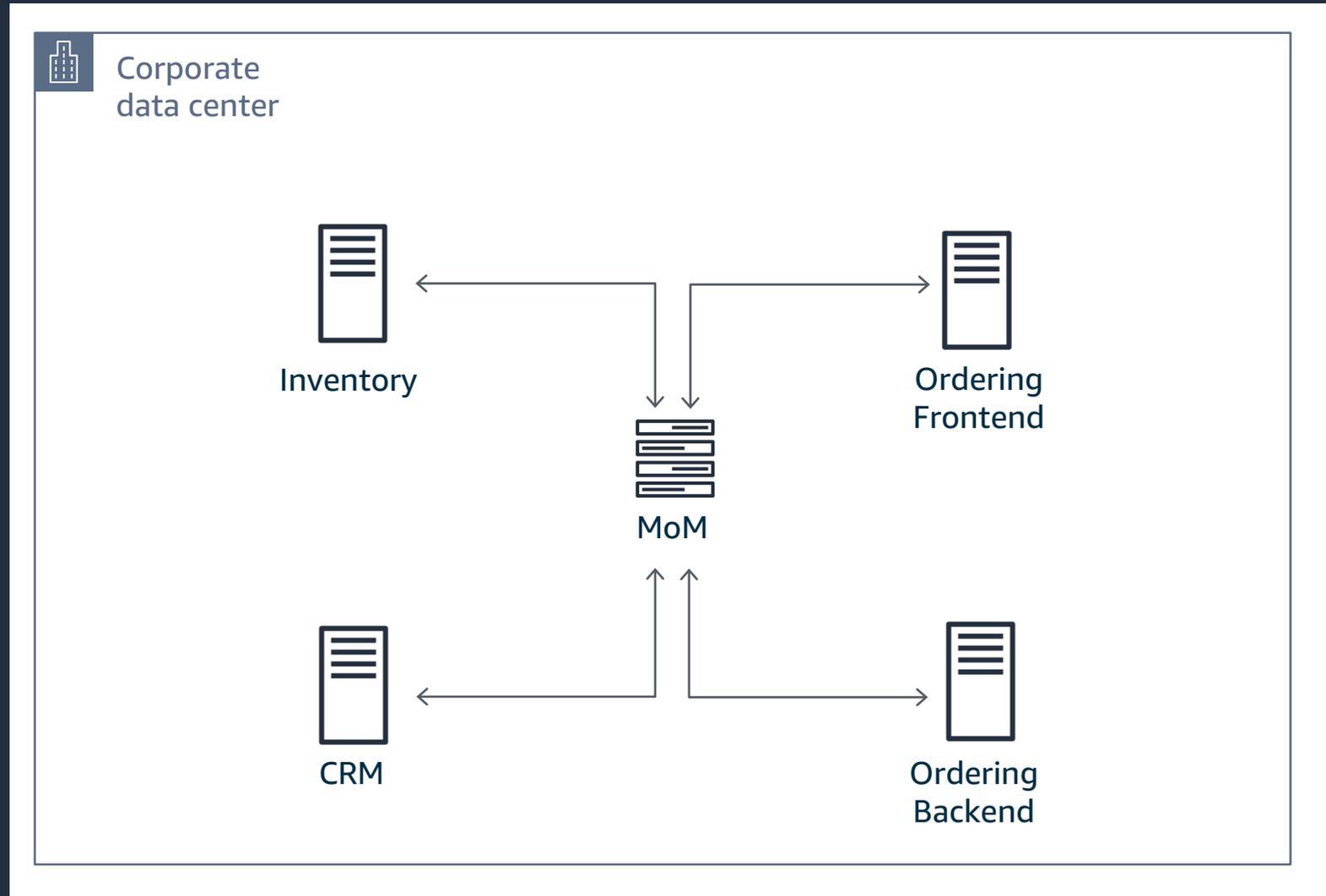


“レガシー” アーキテクチャー

© 2021, Amazon Web Services, Inc. or its Affiliates.

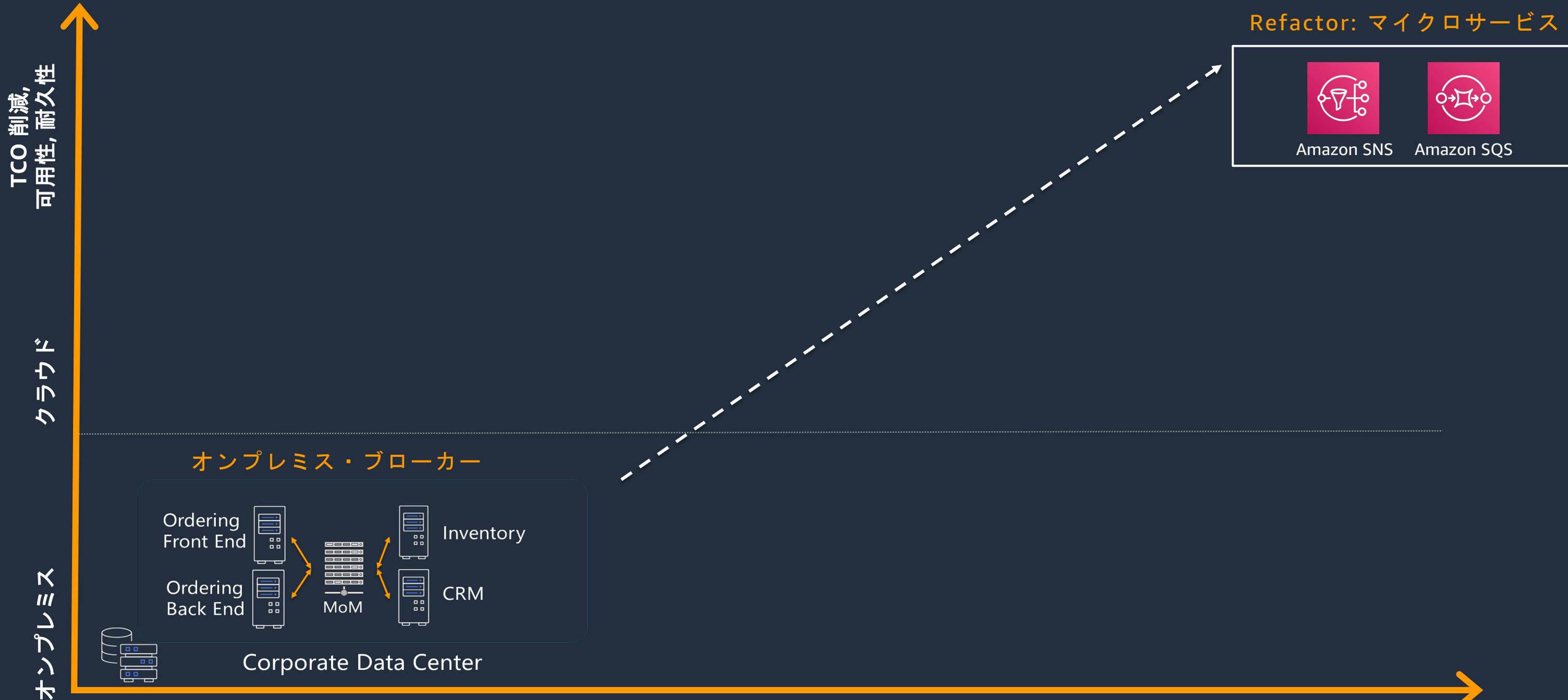
モダナイゼーション 

オンプレミスデータセンターからの移行

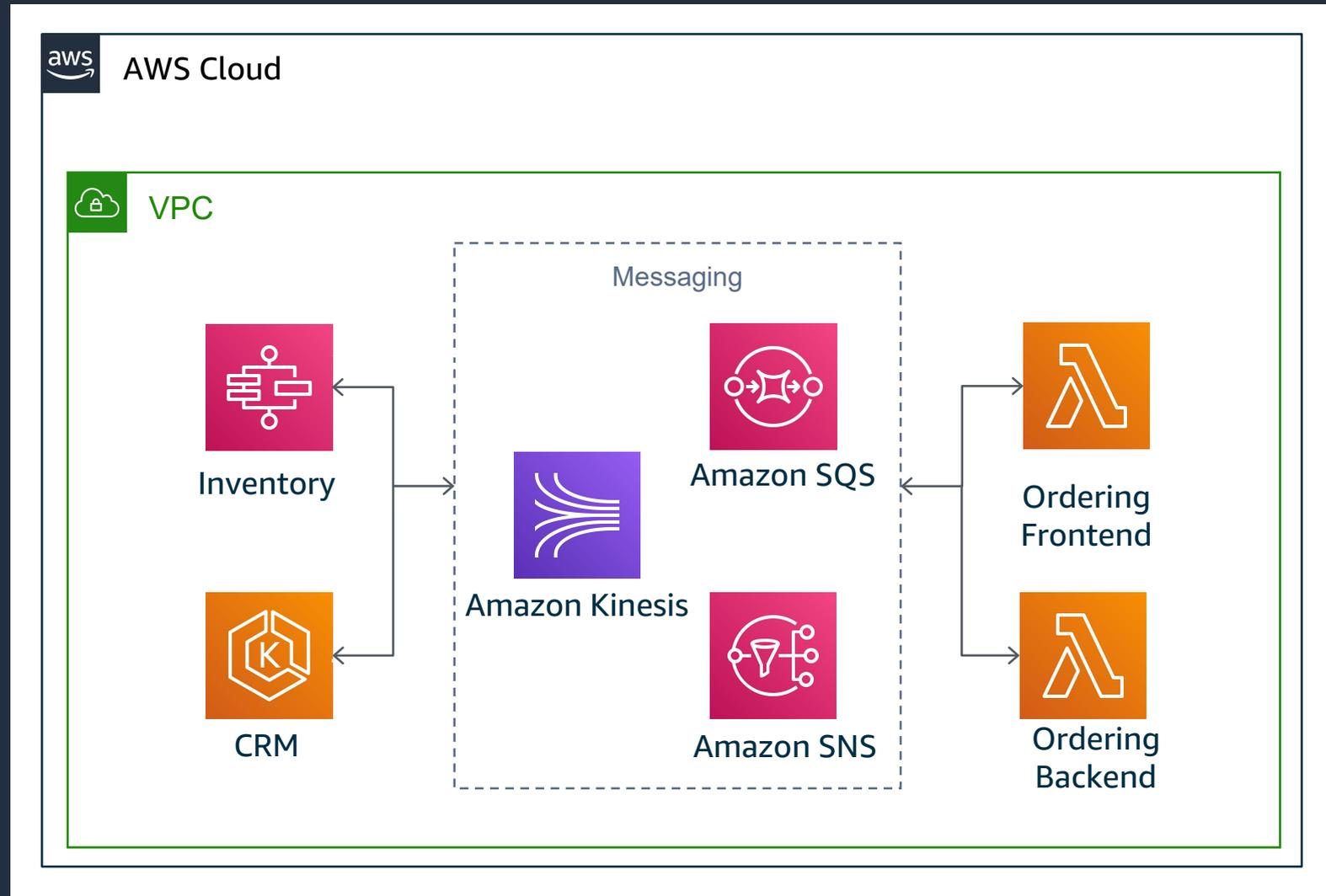


- 長年の運用による蓄積
- ミッションクリティカル
- 高い修正リスク
- 人員、ライセンス、保守に対する高額な投資
- 複雑な管理

アプリケーションの移行パス: Cloud Native



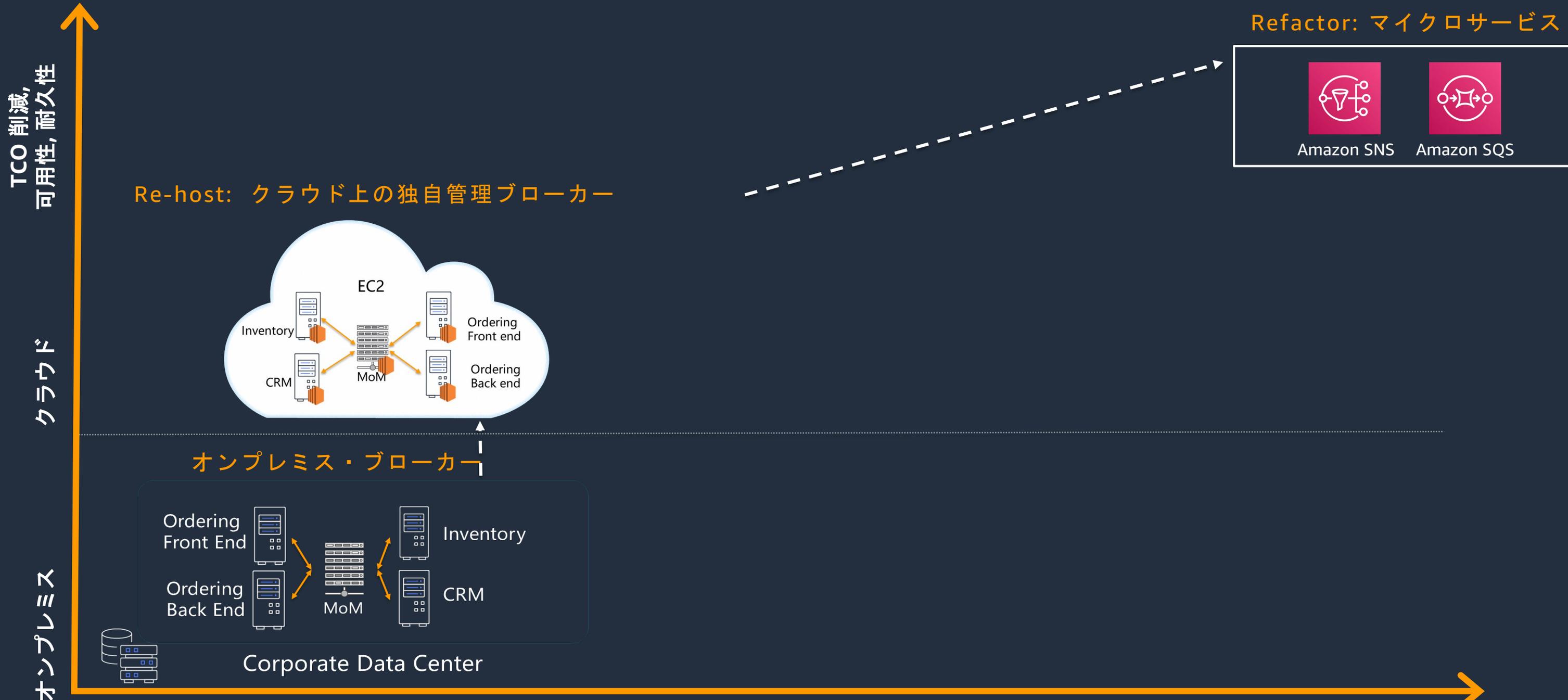
アプリケーション全体のリファクタリング



- + モダンなアーキテクチャ
- + より高い信頼性
- + 最も高いレベルの拡張性
- + 全てをマネージドサービスに

- 高額な開発コスト
- 長期間の開発サイクル
- 最も高いリスク

アプリケーションの移行パス: Lift and shift



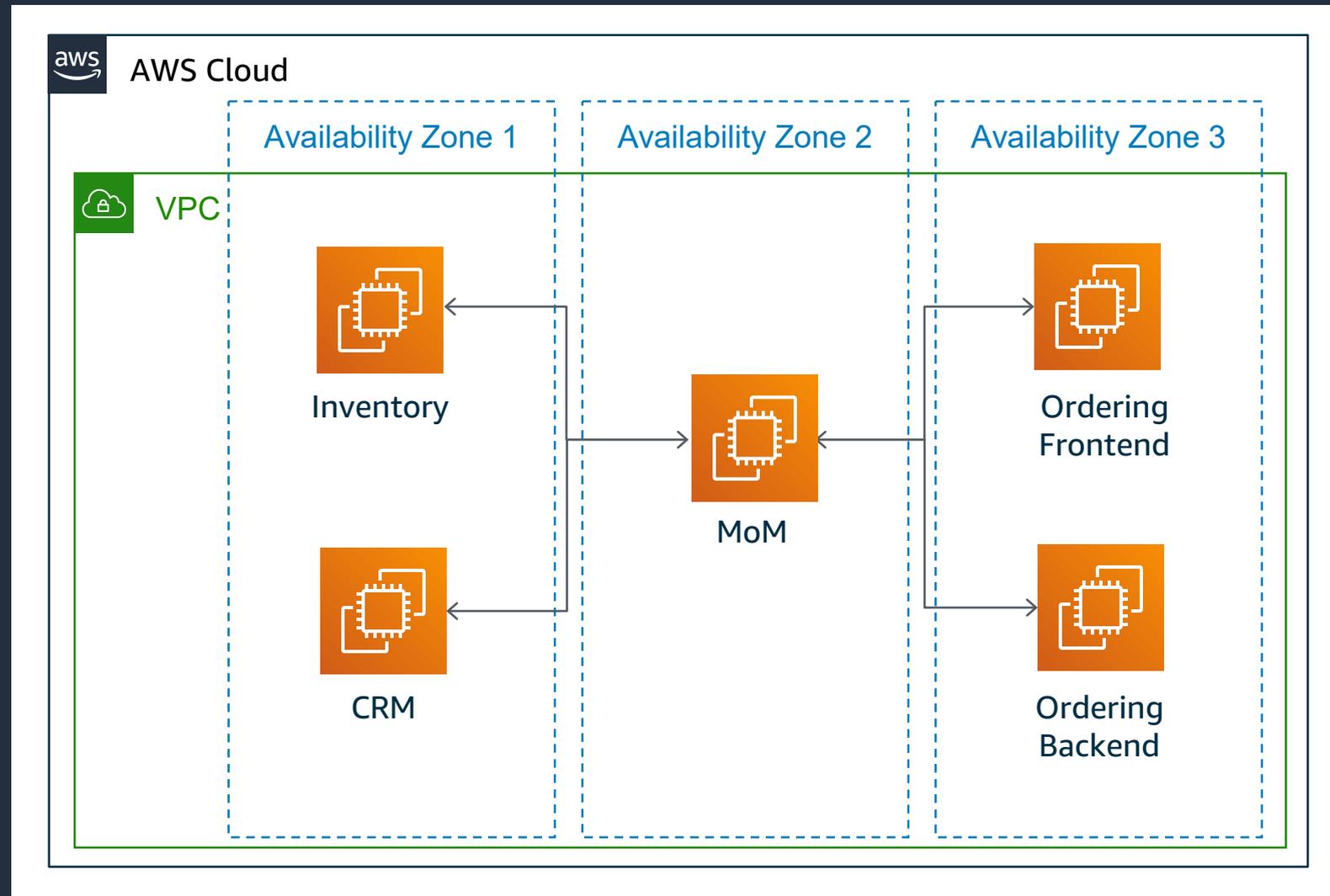
“レガシー” アーキテクチャ

© 2021, Amazon Web Services, Inc. or its Affiliates.

モダナイゼーション

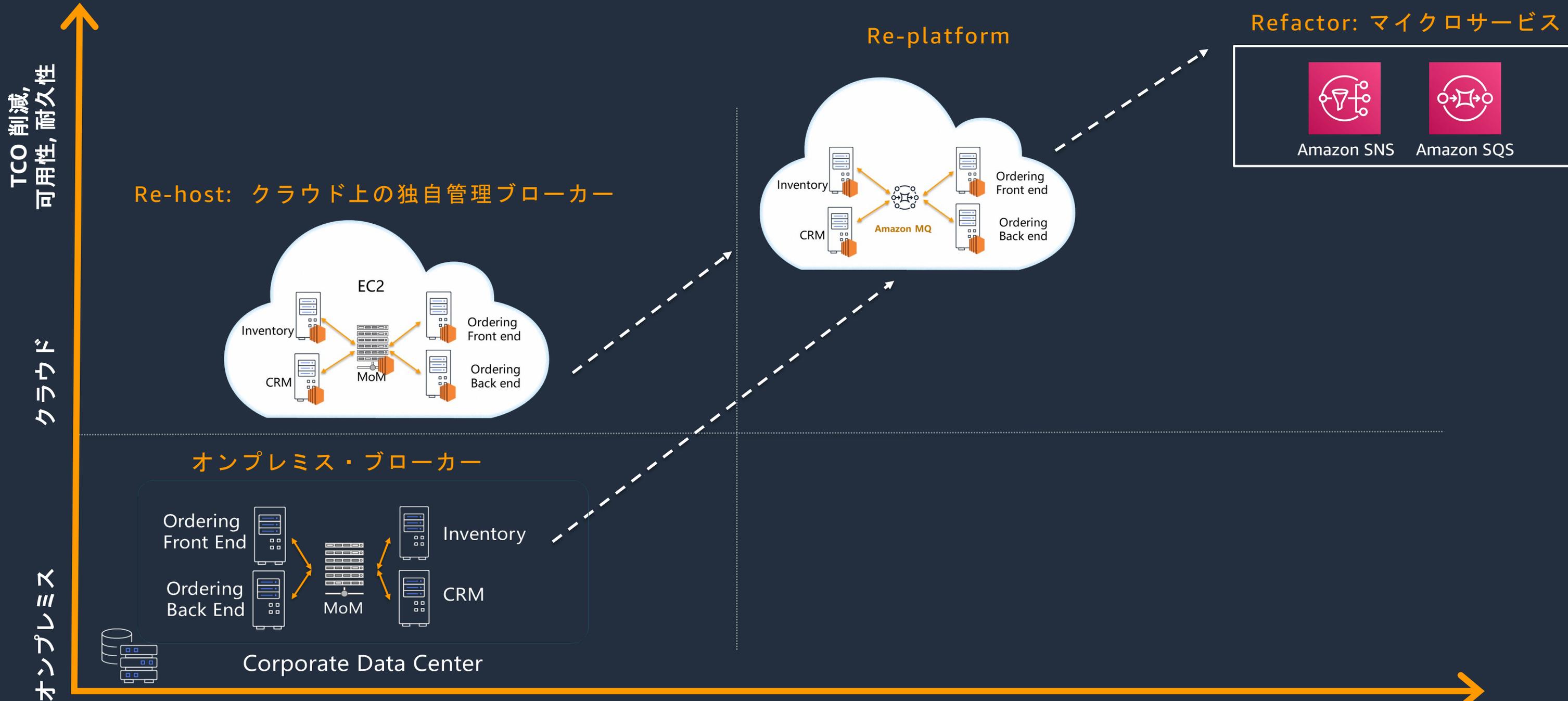


AWS へシステムを Lift and shift



- + インフラコストの軽減
- + より高いセキュリティ
- + より高い可用性
- + より高い柔軟性
- 以前として管理は煩雑
- ライセンスやサポート費用が発生

アプリケーションの移行パス: Amazon MQ



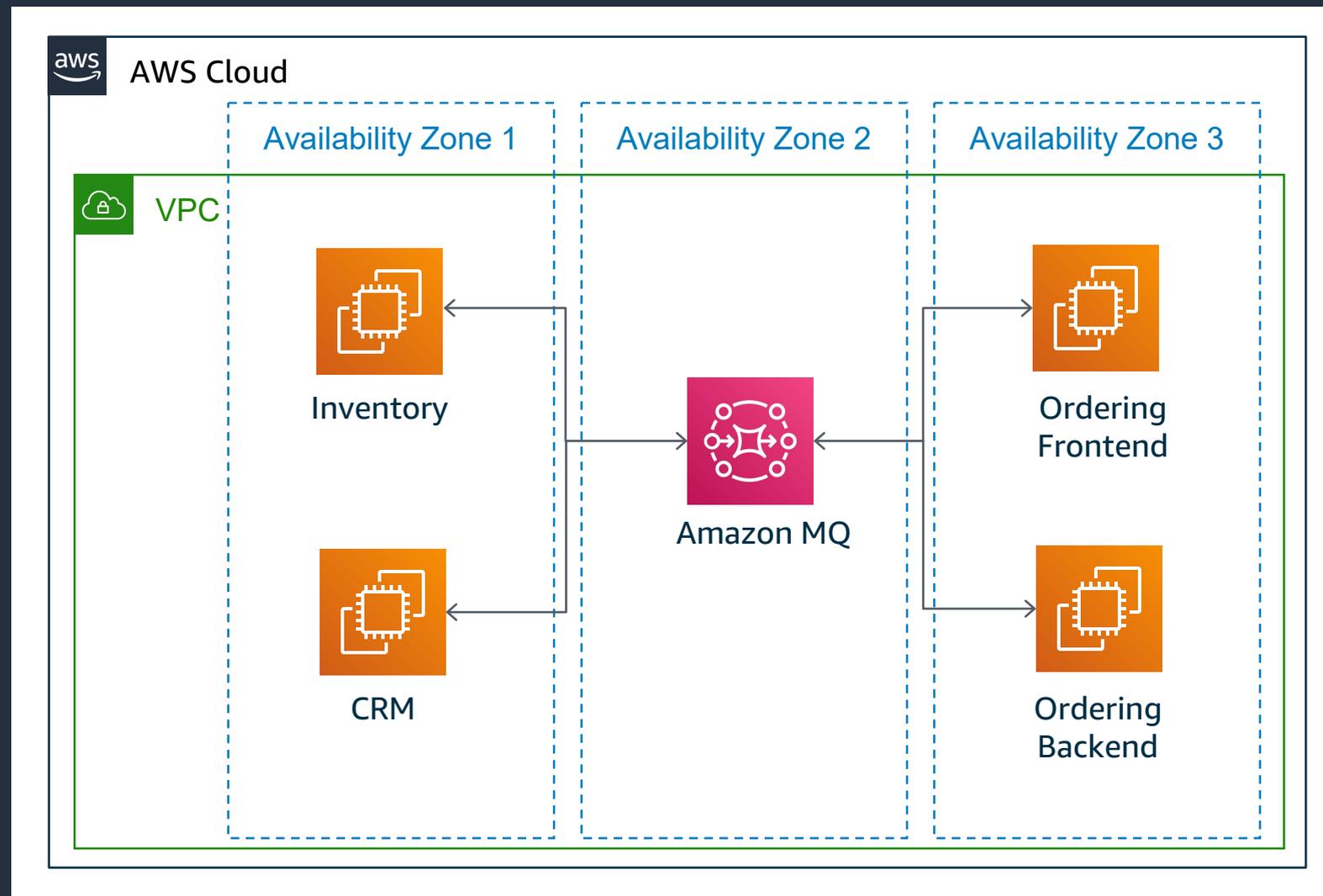
"レガシー" アーキテクチャー

© 2021, Amazon Web Services, Inc. or its Affiliates.

モダナイゼーション



Lift and shift と共にマネージドなブローカーへ置き換え



- + 最小限のコード変更
- + メッセージングシステムの管理をAWSが行う
- + 高い可用性を持つBroker
- + メッセージの永続化
- + より高い信頼性
- + 高額なライセンス

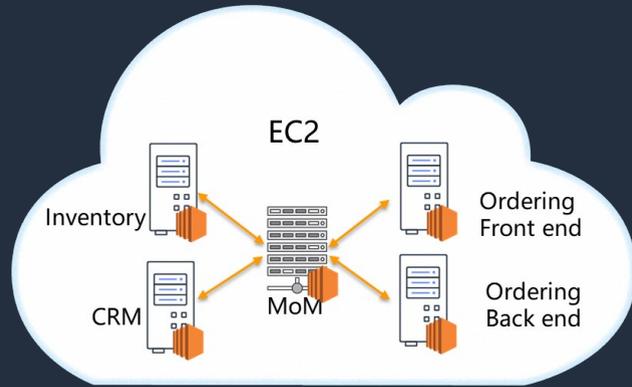
アプリケーションの移行パス: Hybrid & Modernize

TCO 削減,
可用性, 耐久性

クラウド

オンプレミス

Re-host: クラウド上の独自管理ブローカー



オンプレミス・ブローカー



Corporate Data Center

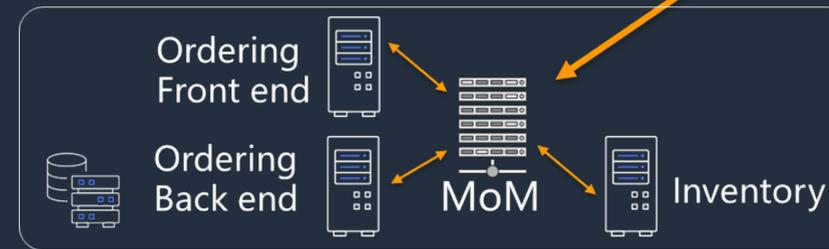
Re-platform



Refactor: マイクロサービス



Hybrid & Modernize



Premises data center

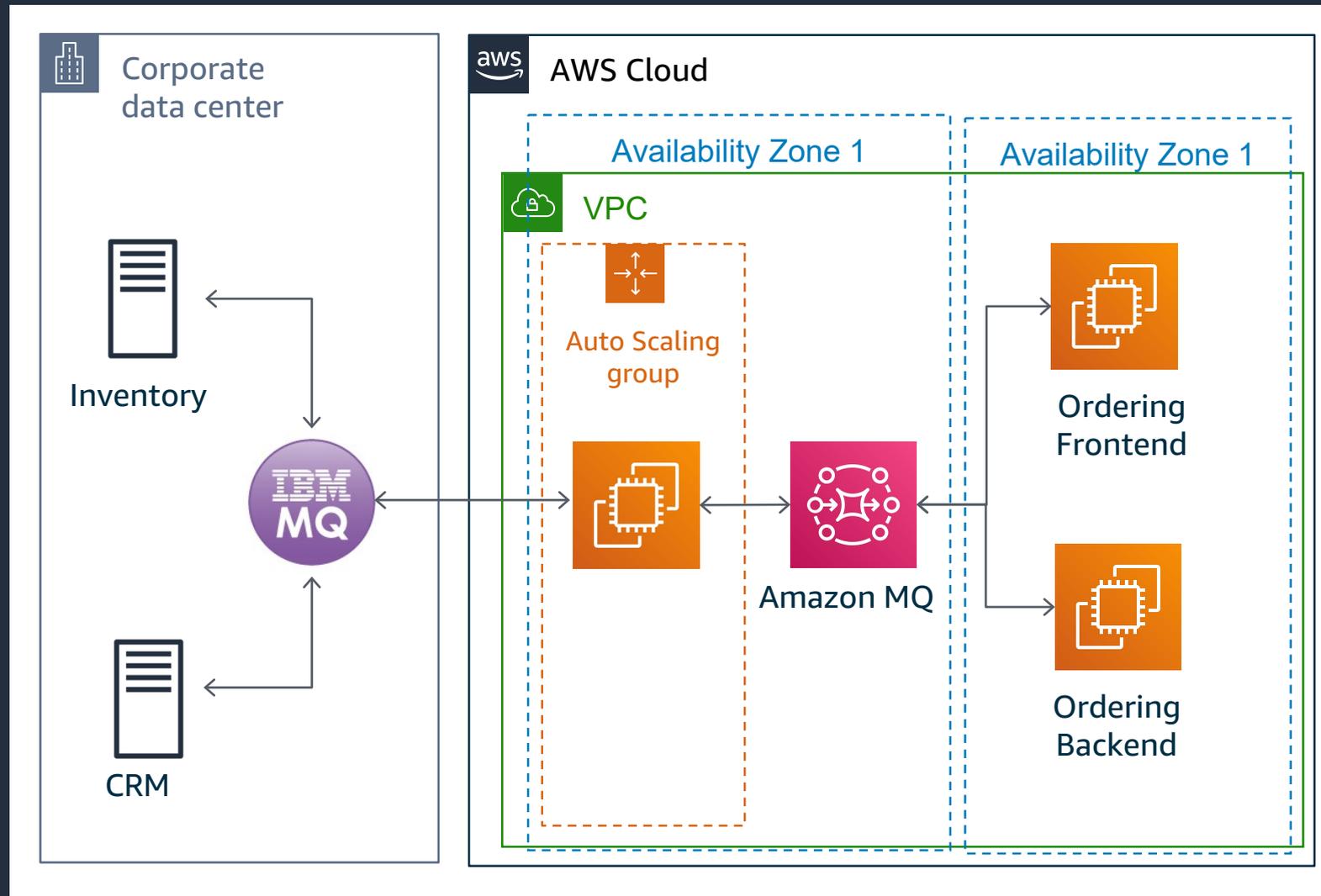
“レガシー”アーキテクチャ

© 2021, Amazon Web Services, Inc. or its Affiliates.

モダナイゼーション



レガシーとモダンを繋ぐ Hybrid solution



- + 五月雨式のアプリケーション移行
- + 新規システムはAWSが管理
- + より高い信頼性
- + 新たなアプリをクラウドに作成
- + オンプレミスのシステムと関係

<https://aws.amazon.com/blogs/compute/migrating-from-ibm-mq-to-amazon-mq-using-a-phased-approach/>

ワークロードに適した移行パターンを検討する

手段 (現行の実装) と目的 (ビジネス上の要求) の混同に注意する

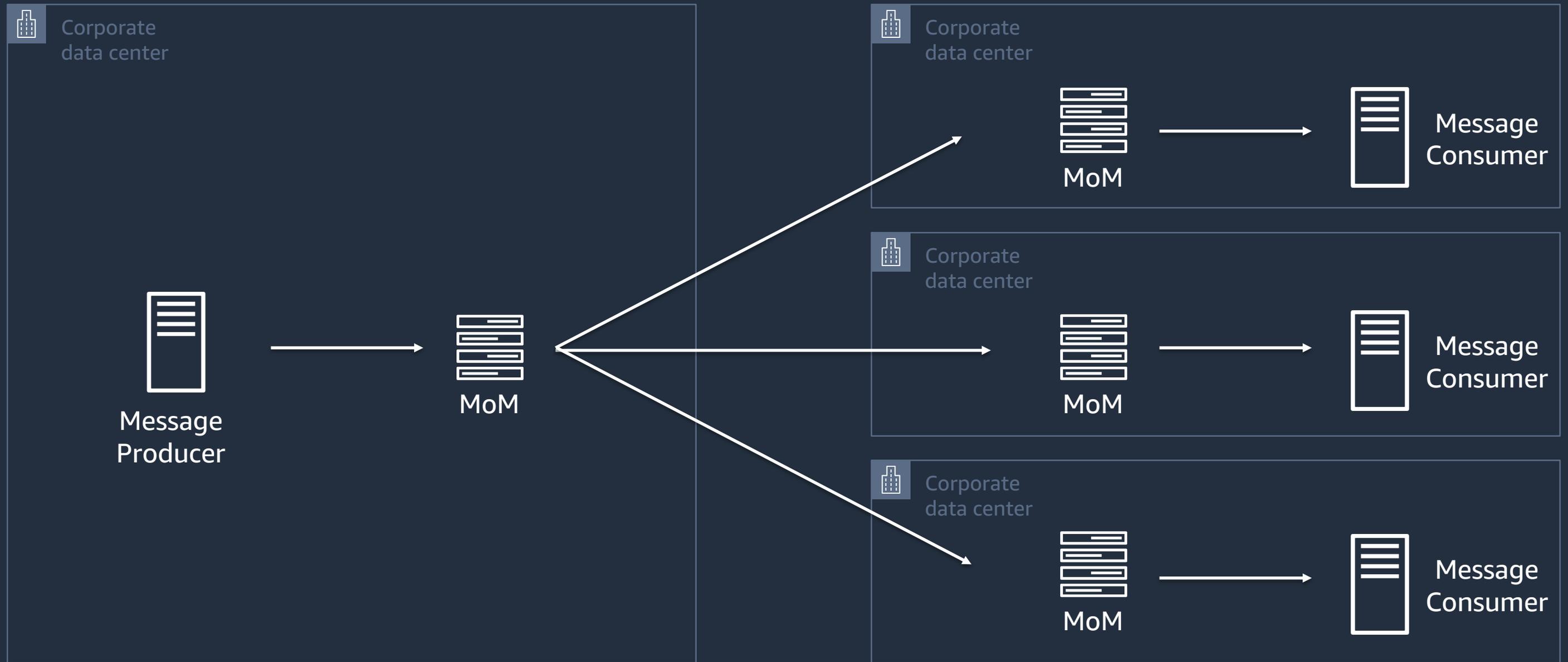
最適なテクノロジーを選定するためには、ビジネス要求に立ち戻り必要な仕様を再精査することが有効です

- 配信ポリシー (at-least-once, exactly-once, etc.)
- メッセージの順序制御
- メッセージの到達時間
- メッセージの耐久性
- ブローカーの可用性
- スケーラビリティ (トランザクション量, クライアント数)
- クライアント言語 (Java, .Net, C++, C#, etc.)
- コスト (初期, 運用時)



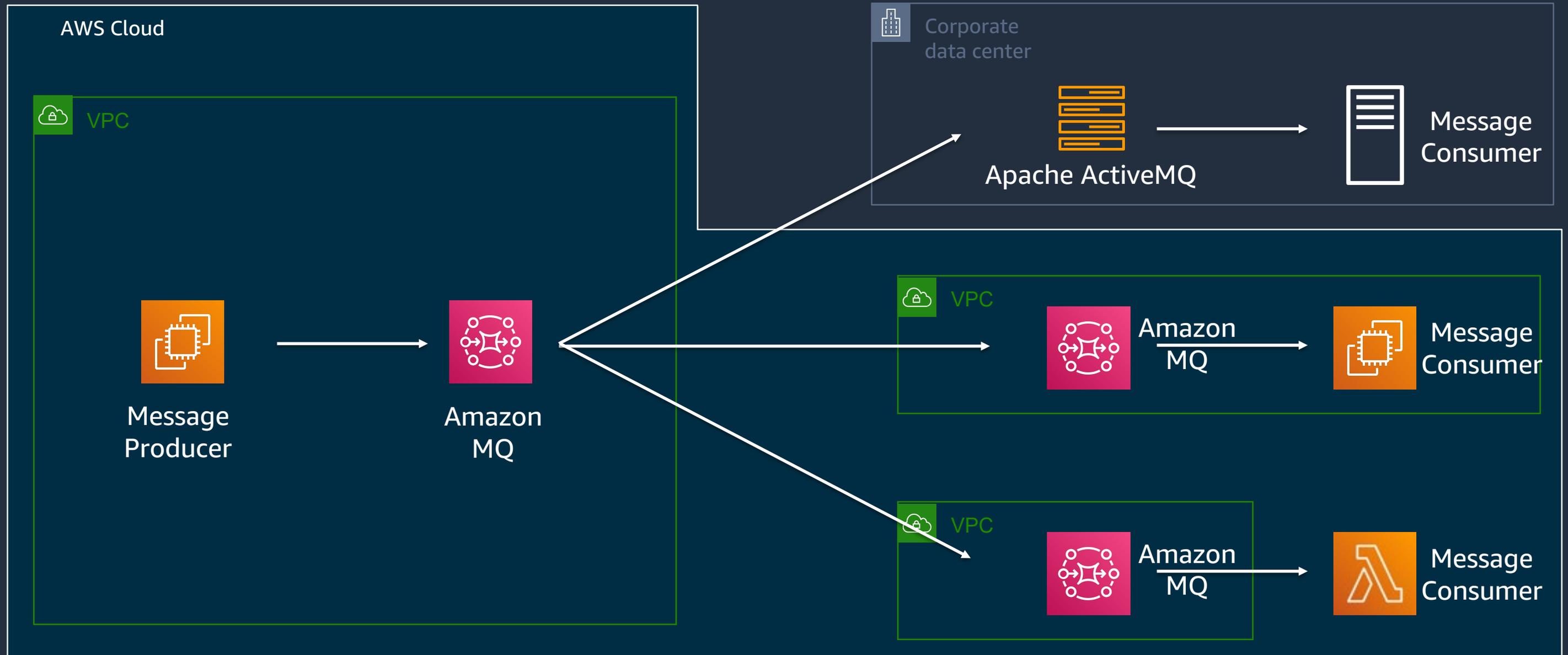
移行パターンの検討例

AS-IS



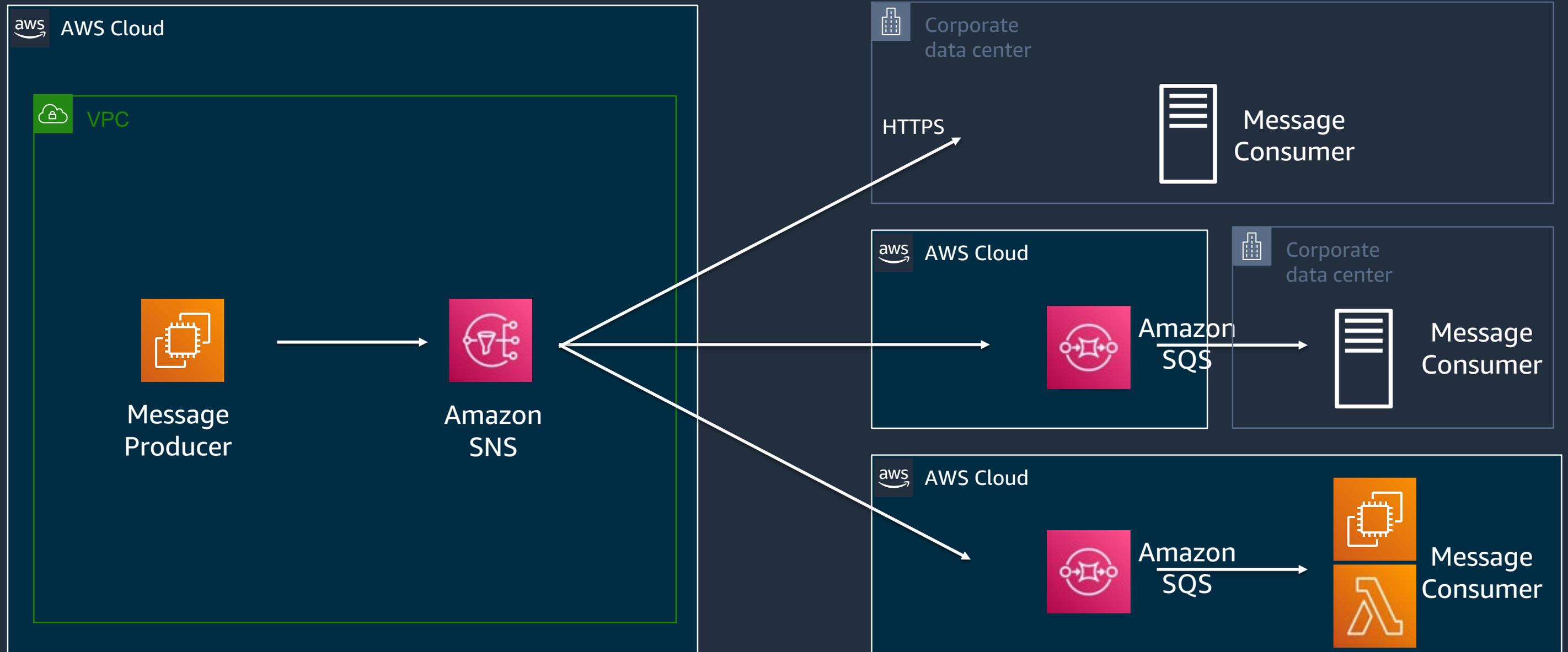
移行パターンの検討例

Amazon MQ + Active MQ



移行パターンの検討例

Amazon SNS + Amazon SQS



本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ for ActiveMQ
- Amazon MQ for RabbitMQ
- Amazon MQ エンジン共通の要素
- Amazon MQへの移行
- その他



リージョン

Amazon MQ は東京リージョンを含む以下のリージョンで利用可能

- バージニア北部
- オハイオ
- オレゴン
- アイルランド
- フランクフルト
- シンガポール
- シドニー
- ソウル
- 東京
- シンガポール
- ロンドン
- モントリオール
- ムンバイ
- パリ
- ストックホルム
- サンパウロ
- 香港
- バーレーン
- ミラノ

サービスレベルアグリーメント (SLA)

- アクティブ/スタンバイ、およびクラスター メッセージブローカーに対して提供される
- 99.9% の月間稼働率で利用可能とするように商業上合理的な努力を行う
- Amazon MQ がサービスコミットメントを満たさない場合、サービスクレジットを受け取る資格が発生する
- 現在 Amazon MQ を利用できるすべてのリージョンを対象とする

Amazon MQ で 99.9% のサービスレベルアグリーメント (SLA) を発表

<https://aws.amazon.com/jp/about-aws/whats-new/2019/01/amazon-mq-announces-service-level-agreement/>

Amazon MQ Service Level Agreement

<https://aws.amazon.com/jp/amazon-mq/sla/>

Amazon MQ の主な制限

ブローカー

- ブローカー数 (リージョンあたり) : **20**
- ブローカー設定履歴の深さ: **10**
- ワイヤレベルプロトコルあたりの接続数 (RabbitMQは適用外) : **1,000** (mq.t2.micro の場合: **100**)
- ブローカーあたりのセキュリティグループ数: **5**
- CloudWatchでモニタリングされる送信先
 - ActiveMQ: **最初の 200 (キューとトピック)**
 - RabbitMQ: **Consumer数の上位 500 (キュー)**

データストレージ

- ブローカーあたりのストレージ容量: **200 GB** (mq.t2.micro ブローカーの場合: **20 GB**)

Amazon MQ API スロットリング

- バケットサイズ: **100**, 補充レート: **15/秒** (ブローカーメッセージングAPIは対象外)

その他

- 名前やパスワードに使用できる文字種の制限等

※ 最新の情報はドキュメントをご確認ください。

Amazon MQ の制限: https://docs.aws.amazon.com/ja_jp/amazon-mq/latest/developer-guide/amazon-mq-limits.html

料金

ブローカーインスタンスの料金

- メッセージブローカーの使用量に基づいて時間単位で (1 秒単位の粒度で) 課金
- 料金は、メッセージブローカーインスタンスのサイズ、選択する構成 (単一、アクティブ/スタンバイ、またはクラスター) による
- mq.m5.large の ActiveMQ アクティブ/スタンバイブローカーの場合: **1時間あたり 0.744USD**

ブローカーストレージの料金

- 月単位での平均ストレージ使用量に基づいた課金
- Amazon EFS の場合、**GB-Month あたり 0.36USD**
- Amazon EBS の場合、**GB-Month あたり 0.12USD**

データ転送料金

- Amazon MQ で送受信されたデータに対して通常の AWS データ転送料金が発生

AWS 無料利用枠

- 1 年間にわたり毎月、単一インスタンスの mq.t2.micro または mq.t3.micro ブローカーを 1 か月あたり最大 750 時間ご利用いただけます。
※ ストレージとして 5GB の EFS (ActiveMQ) または 20GB の EBS (RabbitMQ) を含む

まとめ：Amazon MQ の主な利点

- 最小限のコードの変更で、オーバーヘッドとコストを削減
 - レガシーなブローカーを管理するためのオーバーヘッドが不要に
 - 高価な商用ライセンスが不要に
- クラウド移行とアプリケーションのモダナイゼーションを加速
 - オンプレミスとクラウドを接続し、ハイブリッドメッセージングアーキテクチャを実現
 - 個別のアプリケーションがAWS利点を活用できるよう移行とリファクタを実施
- フル機能かつセキュア、高耐久性、高可用性
 - 必要に応じて即時にメッセージブローカーをデプロイ
 - AWSサービスとの統合

参考情報

Amazon MQ 公式サイト: <https://aws.amazon.com/amazon-mq/>

Amazon Web Services ブログ

- Amazon MQ – ActiveMQのマネージドメッセージブローカーサービス
<https://aws.amazon.com/jp/blogs/news/amazon-mq-managed-message-broker-service-for-activemq/>
- Amazon MQ アップデート — 新しい RabbitMQ メッセージブローカーサービス
<https://aws.amazon.com/jp/blogs/news/amazon-mq-update-new-rabbitmq-message-broker-service/>

Amazon MQ Forum: <https://forums.aws.amazon.com/forum.jspa?forumID=279>

Amazon MQ Twitch: <https://www.twitch.tv/videos/259845421>

参考情報：ハンズオン

- Message-based Integration with Amazon MQ for RabbitMQ (RabbitMQ)
<https://app-int-with-amazonmq.workshop.aws/>
- Amazon MQ Workshop (ActiveMQ)
<https://github.com/aws-samples/amazon-mq-workshop>

参考情報：移行について

- Migrating from RabbitMQ to Amazon MQ
<https://aws.amazon.com/blogs/compute/migrating-from-rabbitmq-to-amazon-mq/>
- Migrating message driven applications to Amazon MQ for RabbitMQ
<https://aws.amazon.com/blogs/compute/migrating-message-driven-applications-to-amazon-mq-for-rabbitmq/>
- Amazon MQ Update – New RabbitMQ Message Broker Service
<https://aws.amazon.com/blogs/aws/amazon-mq-update-new-rabbitmq-message-broker-service/>
- Migrating from IBM MQ to Amazon MQ using a phased approach
<https://aws.amazon.com/blogs/compute/migrating-from-ibm-mq-to-amazon-mq-using-a-phased-approach/>

Q&A

お答えできなかったご質問については

AWS Japan Blog <https://aws.amazon.com/jp/blogs/news/> に
後日掲載します。

AWS の日本語資料の場所「AWS 資料」で検索



日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼

コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#)

[AWS 初心者向け »](#)

[業種・ソリューション別資料 »](#)

[サービス別資料 »](#)

<https://amzn.to/JPArchive>



AWS Well-Architected 個別技術相談会

毎週“W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に
対策などを相談することも可能

- **申込みはイベント告知サイトから**
(<https://aws.amazon.com/jp/about-aws/events/>)

AWS イベント

で[検索]



4月以降のBlack Belt Online Seminarについて

ライブ配信によるBlack Belt Online Seminarは3月一杯で終了し、
今後はオンデマンドによる定期配信に変更いたします。

今後もコンテンツを拡充して行きますので、楽しみにお待ちしております。

オンデマンドでの配信スケジュールは、AWS Blog, AWSニュースレターでお知らせいたします（5月17日週に再開を予定しています）



ご視聴ありがとうございました

AWS 公式 Webinar
<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>

