



[AWS Black Belt Online Seminar]

Amazon MQ

サービスカットシリーズ

Solutions Architect 石井 陽介
2019/7/3

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



過去資料
<https://amzn.to/JPArchive>



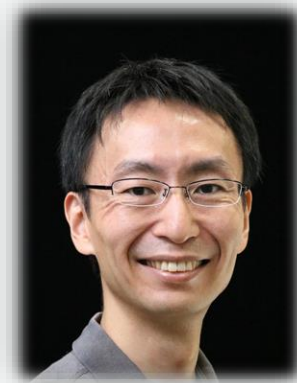
自己紹介

□ 名前

石井 陽介

□ 所属

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のサービス

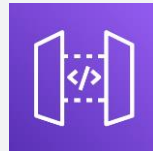
Amazon MQ



AWS Lambda



Amazon API Gateway



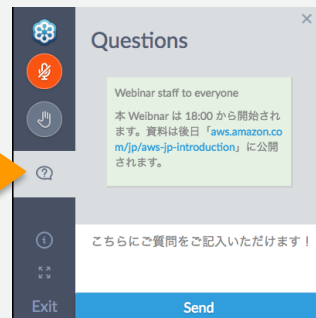
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
お答えできませんのでご了承下さい

- ① 吹き出しをクリック
- ② 質問を入力
- ③ Sendをクリック



Twitter 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2019年7月3日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。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います。日本居住者のお客様が東京リージョンを使用する場合、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 の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- 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 の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- その他



エンタープライズ・アプリケーションを支えるメッセージング



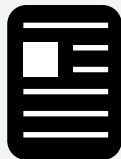
小売業のオーダー処理

- フロントエンドとバックエンド、在庫と請求、マーケティングを接続
- Webサイトの遅延を防止
- 受注情報の欠損を防止



銀行の金融トランザクション

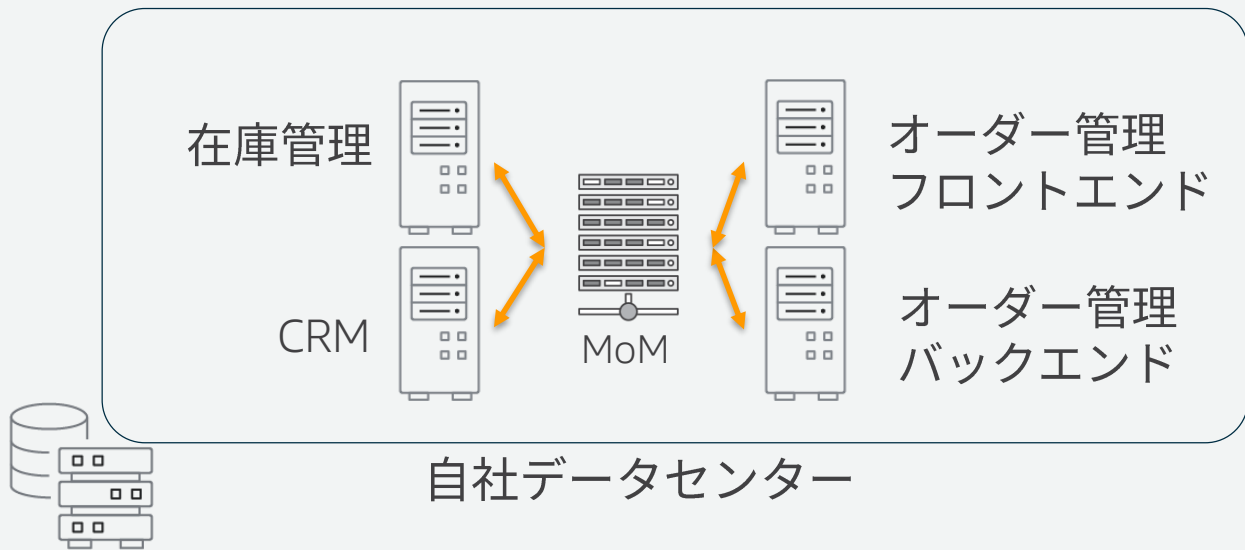
- すべての取引や決済を取り込み
- 各ステップが失敗なく処理されることを保証 (記録保持、分析)
- サードパーティへの情報の配布 (転送)



メディア企業のコンテンツ公開

- 公開のすべき文書の取り込み
- 各ステップを独立して処理 (再フォーマット, 翻訳)
- 文書を多様な送信先へ発信 (web, 印刷, 第三者)

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

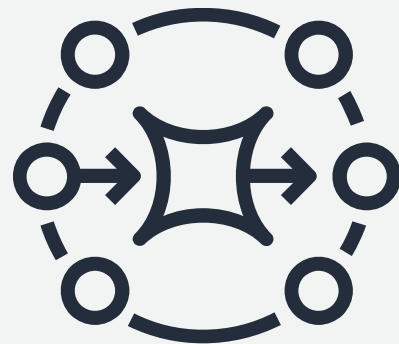
本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- その他



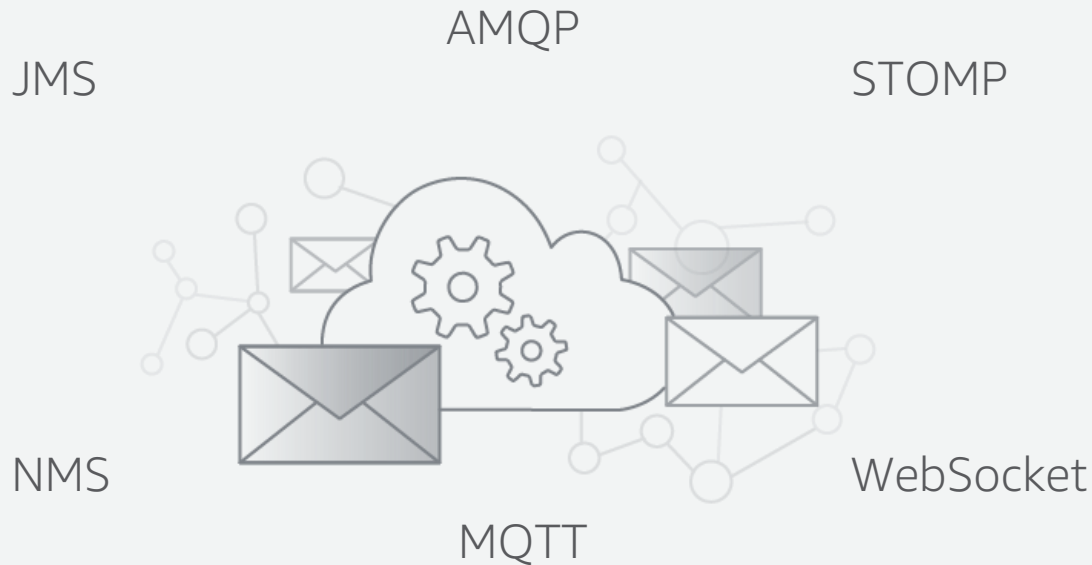
Amazon MQ とは

- マネージドメッセージブローカーサービス
- Apache ActiveMQ をサポート
- 豊富なメッセージング機能
- 低遅延かつ高耐久性
- 業界標準のAPIやプロトコルに対応

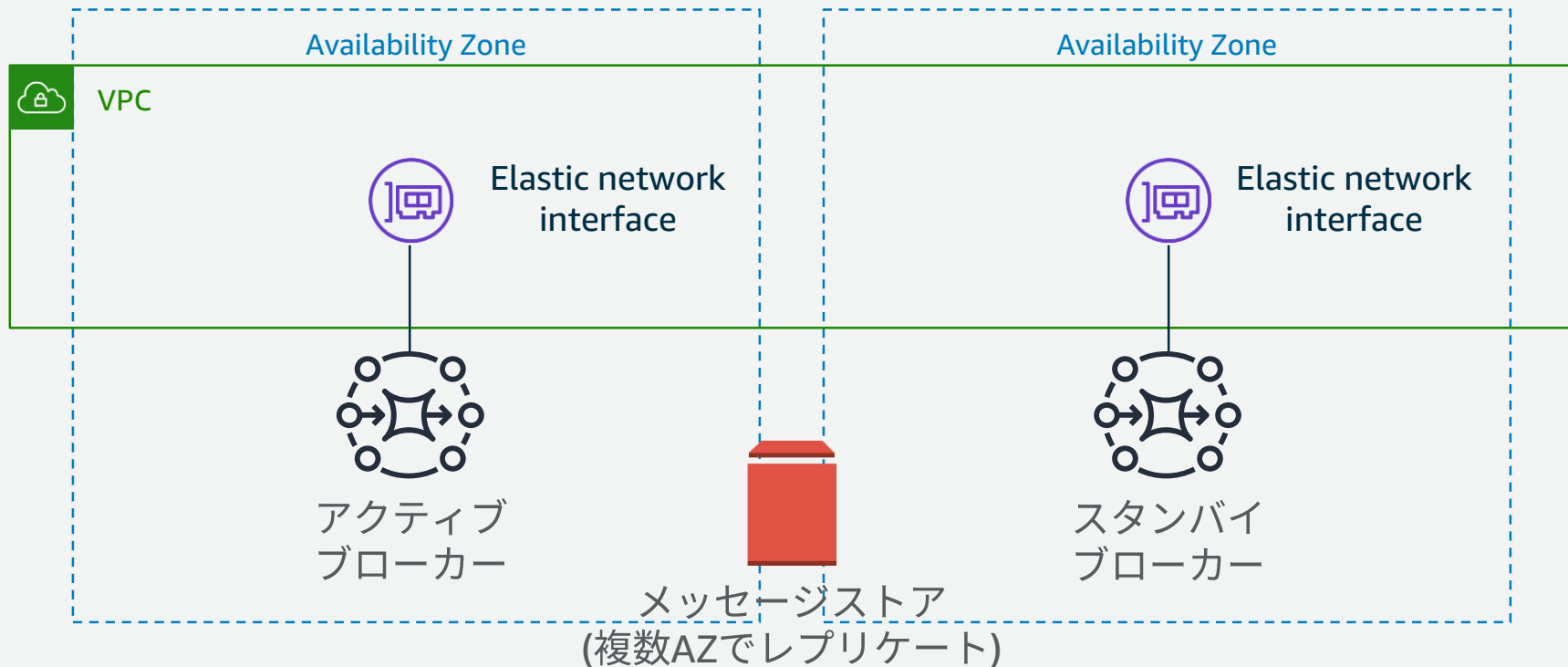


Amazon MQ

少ない変更でアプリケーションを移行



Amazon MQ の機能 – 高可用性と耐久性

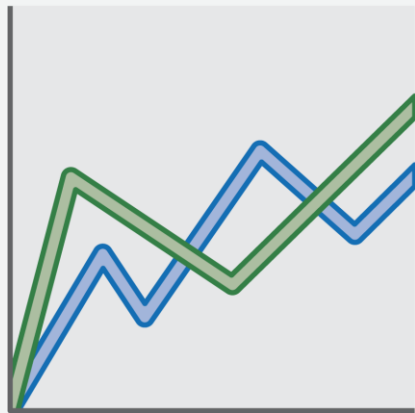


Amazon MQ の機能 – セキュリティ



- 伝送経路は TLS で暗号化される
- 保管データはサービスによって管理されたキーで暗号化される
- ブローカーへの通信をプライベートネットワークに閉じることが可能
- セキュリティグループによるネットワーク防御が可能

Amazon MQ の機能 – Amazon CloudWatch によるモニタリング



- CloudWatch Metrics によってブローカーの稼働状況を取得
 - リソース関連メトリクス
 - キューやトピックのメトリクス
- CloudWatch Alarm を設定可能
- 各種メトリクスに基づいてコンシューマーのフリートに Auto Scaling を適用できる

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- 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

ActiveMQ

Amazon MQ

- 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、セクター 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



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	RabbitMQ	Library	STOMP	Send to Queue, Send to Topic, Receive from Queue, Receive from Topic

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



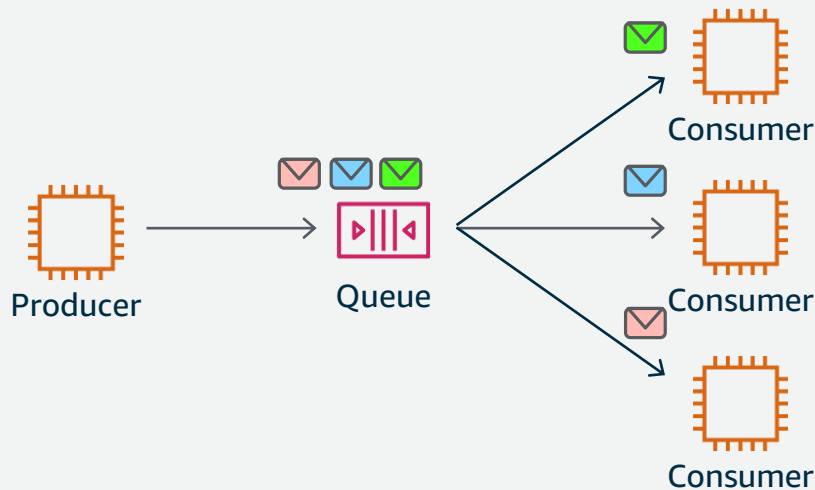
Apache ActiveMQ 概要 – キュー

Point-to-Point で信頼性のあるメッセージングを実現

ActiveMQ

Amazon MQ

- 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



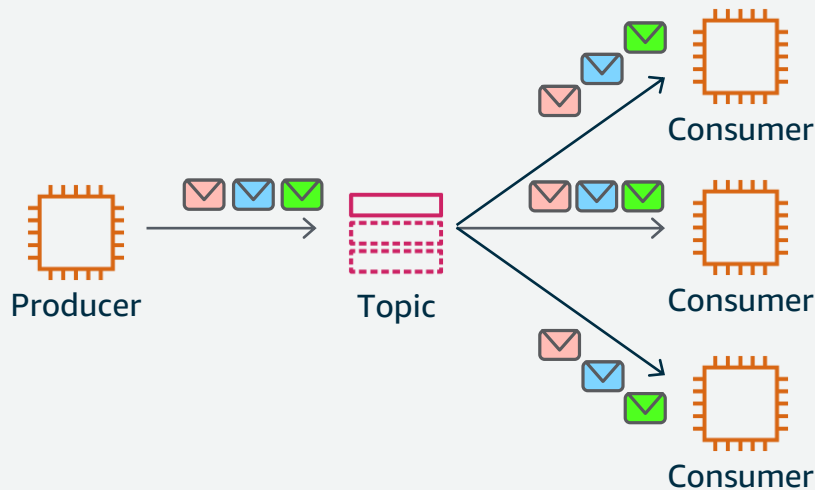
Apache ActiveMQ 概要 – トピック

同一メッセージを複数の送信先へ送達

ActiveMQ

Amazon MQ

- 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



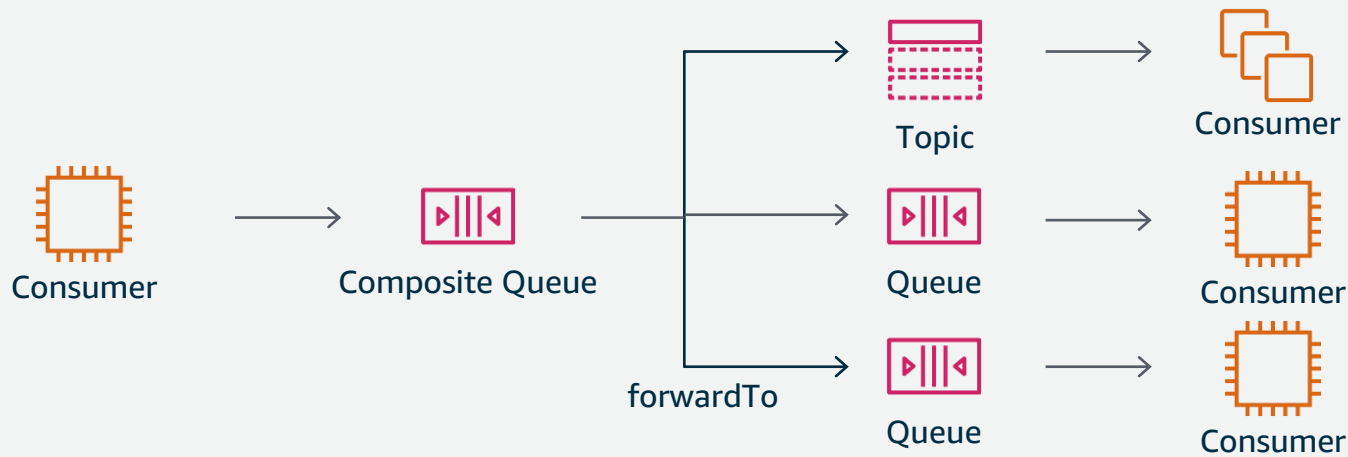
Apache ActiveMQ 概要 – 複合送信先 (Composite Destination)

複数の物理送信先に一括でメッセージをフォワード

ActiveMQ

Amazon MQ

- 1つの仮想送信先に対して複数の物理送信先をマッピング可能
- 物理送信先はトピック/キューのいずれも指定可能



Apache ActiveMQ 概要 – 仮想トピック (Virtual Topic)

トピックの柔軟性とキューの拡張性を両立

ActiveMQ

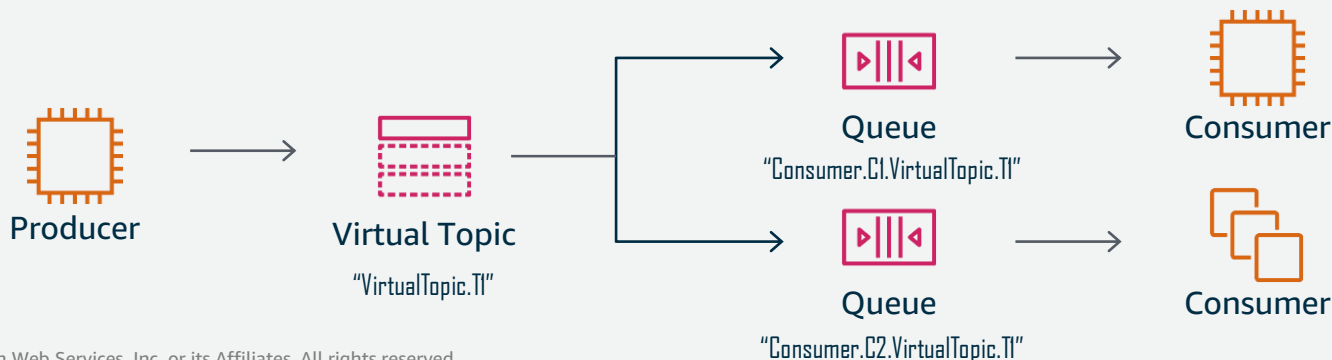
Amazon MQ

- 仮想トピックに送付されたメッセージは対応するキューに転送される
- プロデューサーからは通常のトピック同様に利用可能
- ルーティングは予め決められた命名規約に基づく (事前定義不要)

仮想トピック: `VirtualTopic.<topic-name>`

送信先キュー: `Consumer.<consumer-name>.VirtualTopic.<topic-name>`

- キューは複数のコンシューマーからアクセス可能



Apache ActiveMQ 概要 – 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

ActiveMQ

Amazon MQ

- 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>

Apache ActiveMQ 概要 – スケジュールメッセージ

遅延やスケジュールに基づいたメッセージの配信

ActiveMQ

Amazon MQ

- クライアントから送出されるメッセージの配信タイミングを制御
- 初期遅延、送信間隔、送信回数を指定
- Cron表記によるスケジュール定義にも対応
- ブローカーの schedulerSupport を有効化することで利用可能

例: 毎時0分より、1秒間隔で10回メッセージを送信

```
MessageProducer producer = session.createProducer(destination);
TextMessage message = session.createTextMessage("test msg");
message.setStringProperty(ScheduledMessage.AMQ_SCHEDULED_CRON, "0 * * * *");
message.setLongProperty(ScheduledMessage.AMQ_SCHEDULED_DELAY, 1000);
message.setLongProperty(ScheduledMessage.AMQ_SCHEDULED_PERIOD, 1000);
message.setIntProperty(ScheduledMessage.AMQ_SCHEDULED_REPEAT, 9);
producer.send(message);
```

java

Apache ActiveMQ 概要 – 認可マップ

ユーザーロールによるリソースに対する操作権限を制御

ActiveMQ

Amazon MQ

- キューやトピックに対して、メッセージの読み取り (read)、書き込み (write)、実行時作成 (admin) の権限を与える
- キューやトピックの指定にはワイルドカードを指定可能

```
<authorizationPlugin>
  <map>
    <authorizationMap>
      <authorizationEntries>
        <authorizationEntry admin="admins,activemq-webconsole" read="admins,users,activemq-webconsole"
          write="admins,activemq-webconsole" queue=">" />
        <authorizationEntry admin="admins,activemq-webconsole" read="admins,users,activemq-webconsole"
          write="admins,activemq-webconsole" topic=">" />
      </authorizationEntries>
    </authorizationMap>
  </map>
</authorizationPlugin>
```

activemq.xml

ワイルドカード

- “.”: 階層構造の区切り
- “*”: パス階層上の任意の名前に一致
- “>”: 任意のパスに一致

Apache ActiveMQ 概要 – 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ることで
高度なメッセージング要件を実現

ActiveMQ

Amazon MQ

- 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

ActiveMQ

Amazon MQ

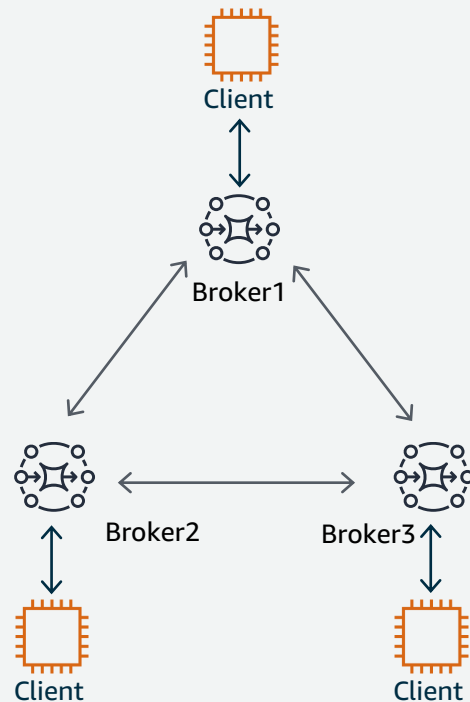
- 接続されたブローカー間で相互にメッセージを転送
- 地理的に分散した拠点間の通信等で使用される

```
<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)



ブローカーのネットワーク - ハブアンドスポークトポロジ

設定や通信パスの集約による、高いメンテナンス性

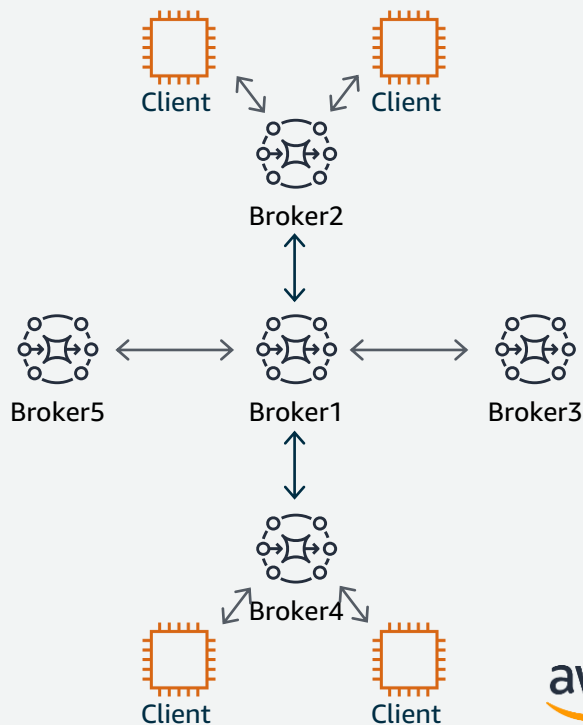
ActiveMQ

Amazon MQ

- Broker1 のネットワークコネクタのみでネットワーク全体を制御可能
- ハブとなるBroker1 に高いサービスレベルが求められる

```
<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)"/>  
  <networkConnector name="con1to4" userName="user" duplex="true"  
    uri="static:(ssl://broker4-hostname:61617)"/>  
  <networkConnector name="con1to5" userName="user" duplex="true"  
    uri="static:(ssl://broker5-hostname:61617)"/>  
</networkConnectors>
```

activemq.xml (Broker1)



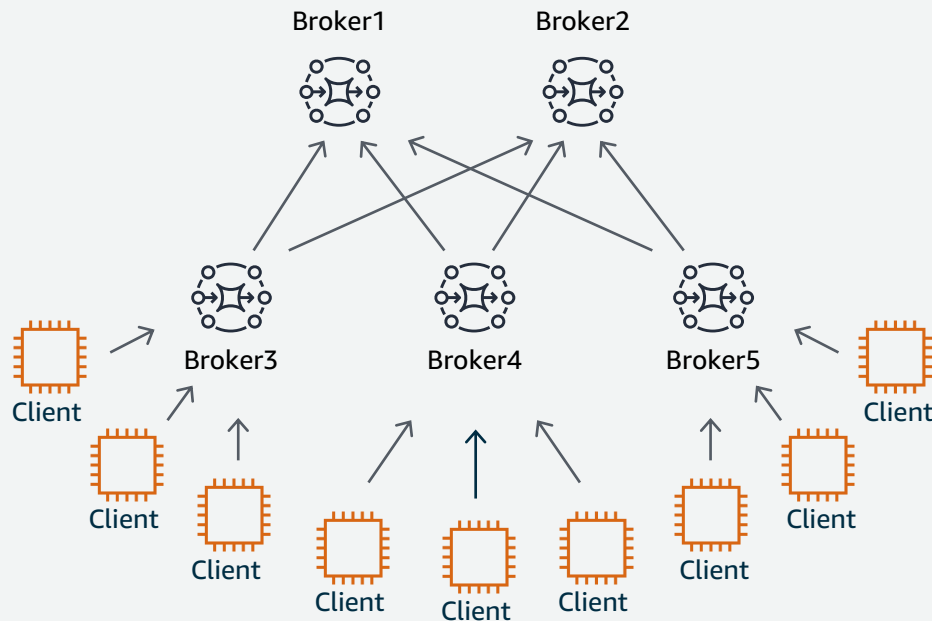
ブローカーのネットワーク - コンセントレータトポロジ

大量のクライアント接続を受け入れることが可能

ActiveMQ

Amazon MQ

- サービス提供するブローカー (上段) に対して、複数のクライアント接続用ブローカー (下段) からメッセージを集約する
- 下段のブローカーを増やすことで、より大量のクライアント接続要求に対応可能

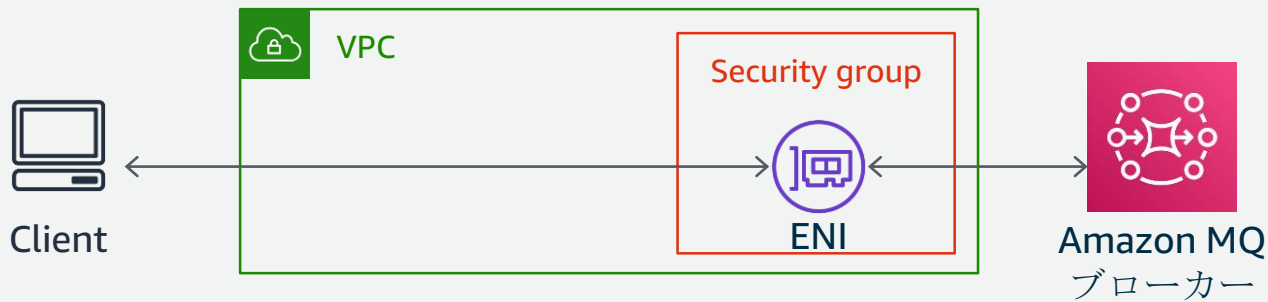


Amazon MQ の構成要素 – ブローカー

Amazon MQ で実行されるメッセージブローカー環境

Amazon MQ

- 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



※ 作成済みのブローカーのネットワーク設定変更するためにはブローカーの再作成が必要

Amazon MQ の構成要素 – ブローカー/エンジン

Amazon MQ

以下のブローカーエンジンをサポート

- Apache ActiveMQ 5.15.9 (推奨)
- Apache ActiveMQ 5.15.8
- Apache ActiveMQ 5.15.6
- Apache ActiveMQ 5.15.0

▼ 追加設定

ブローカーエンジンバージョン

5.15.9 ▼
5.15.9
5.15.8
5.15.6
5.15.0

☒ アップロードした新しい設定を作成する
Amazon MQ は、ブローカーのエンジンと

☐ 既存の設定を選択する

Amazon MQ の構成要素 – ブローカー/インスタンスタイプ

Amazon MQ

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.t2.micro	1	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.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。

※ 作成済みのブローカーのインスタンスタイプ変更するためにはブローカーの再作成が必要

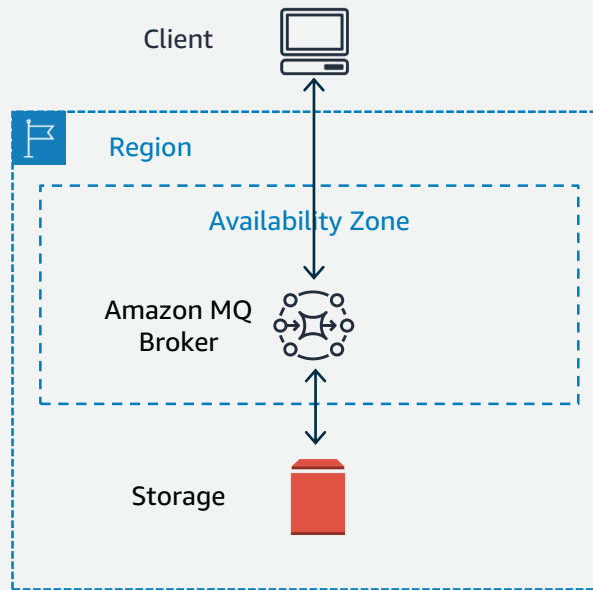
単一インスタンスブローカー

Amazon MQ

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

※ 作成済みのブローカーのデプロイモード変更するためにはブローカーの再作成が必要

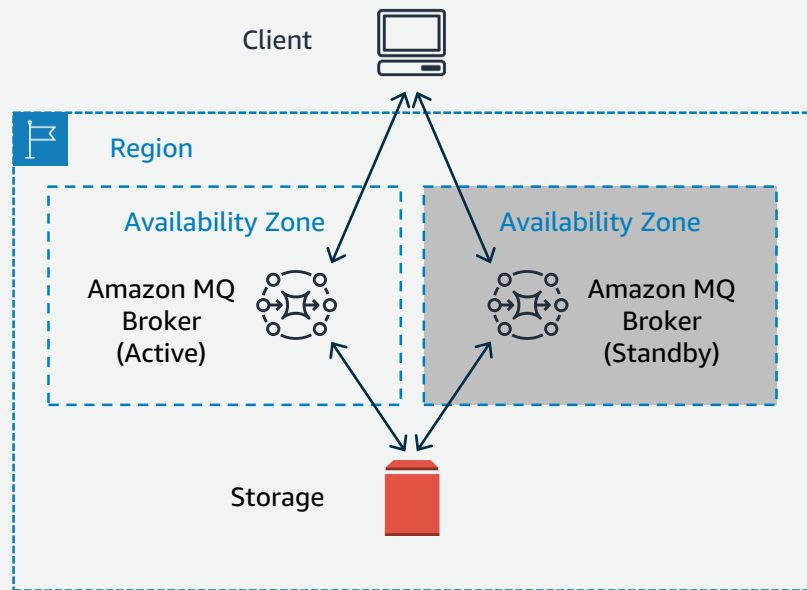


アクティブ/スタンバイブローカー

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

Amazon MQ

- 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



Amazon MQ の構成要素 – ActiveMQ ウェブコンソール

ActiveMQ の設定を行う Web コンソール

Amazon MQ

- ブローカーの接続情報からURLを確認可能
- アクティブ/スタンバイブローカーでは、アクティブ側のエンドポイントのみアクセス可能
- 初期ログインユーザーはブローカー作成時に指定
- ブローカーのパブリックアクセスを許可しない場合、別途踏み台サーバーやプロキシ、VPN等によってアクセス経路を確保する必要がある



ActiveMQ ウェブコンソール URL

`https://<broker-id>-<instance-number>.mq.<region>.amazonaws.com:8162`

<broker-id>: ブローカーのARNより確認

<instance-number>: 単一インスタンスブローカーでは1
アクティブ/スタンバイブローカーではそれぞれ1または2

Amazon MQ の構成要素 – 設定

Amazon MQ ブローカーの挙動を定義する XML ファイル

Amazon MQ

- 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

※ 現在、一度作成された設定を削除することはできない

```
<?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)

Amazon MQ

plugins

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

destinationInterceptors

- mirroredQueue
- virtualTopic
- compositeQueue

destinationPolicy

- pendingMessageLimitStrategy
- dispatchPolicy
- subscriptionRecoveryPolicy

destinations

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 の構成要素 – ユーザー

ブローカーのキューやトピックへアクセスする主体を定義

Amazon MQ

- ActiveMQ の Simple Authentication Plugin を使用
- AWS マネジメントコンソールからユーザーとグループを定義
- ユーザー名とパスワードによる認証



The screenshot shows the 'Create User' dialog in the Amazon MQ console. It includes fields for 'User Name' (filled with 'bb-user'), 'Password' (masked with dots), and 'Group - Options' (filled with 'admins, users'). There is a checkbox for 'ActiveMQ Web Console' which is checked. Below the password field, there is a note: '12 文字以上で、一意の文字を 4 つ以上含める必要があります。' (Must be 12 characters or more, and contain 4 or more unique characters). Below the group field, there is a note: 'グループのカンマ区切りの一覧' (A comma-separated list of groups).

ユーザーの作成

ユーザー名

bb-user

カンマ (,), コロン (:), 等号 (=), またはスペースを含めることはできません。

パスワード

12 文字以上で、一意の文字を 4 つ以上含める必要があります。

☐ 表示

グループ - オプション

admins, users

グループのカンマ区切りの一覧

☒ ActiveMQ ウェブコンソール

※ 現在、クライアント証明書の認証や、JAAS (Java 認証および許可サービス) 用のプラグインは非サポート

Amazon MQ の構成要素 – ライフサイクル

設定やユーザーへの変更の反映にはブローカーのリブートが必要

Amazon MQ

- 設定リビジョンや ActiveMQ ユーザーへの変更は即座には適用されない
- 次回のスケジュールされたメンテナンスウィンドウでの適用か、即時適用するかを選択
- 適用はブローカーのリブートを伴う
- メンテナンスウィンドウを選択した場合も、ウィンドウまでにリブートが発生すると変更が適用される

① これらの変更を適用するには、次のメンテナンス時間を待つか、ブローカーを再起動する必要があります。 [詳細はこちら](#)

メンテナンス

自動マイナーバージョンアップグレード

はい

メンテナンスウィンドウ

日曜日 09:00 - 11:00 UTC

保留中の変更

ブローカー設定

[現行の設定](#)

[保留中の設定](#)

ユーザー

1 ユーザーを追加しました

メンテナンス

ブローカーのメンテナンスに関するポリシーを指定

Amazon MQ

- 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

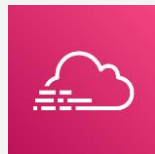
開始日	開始時刻	所要時間
日曜日 ▼	00 ▼ : 00 ▼ UTC	2 時間

※ メンテナンスウィンドウを変更するためにはブローカーの再作成が必要

モニタリング



Amazon CloudWatch



AWS CloudTrail

Amazon MQ

- Amazon CloudWatch

- キュー/トピックのメトリクス
- ブローカー：HeapUsage, CpuUtilization, NetIn, NetOut, TotalMessageCount
- トリガーや通知をサポート

- AWS CloudTrail

- Amazon MQ コンソールや Amazon MQ API を含む、Amazon MQ ブローカーと設定に関連する API コールをイベントとしてキャプチャ

- Apache ActiveMQ Advisory Topics

- ブローカー上でライフサイクルに関する通知を取得するための特別なトピック

- Apache ActiveMQ 統計プラグイン

- ブローカーの統計を取得
- 送信先の統計を取得

※ JMXはサポートされません

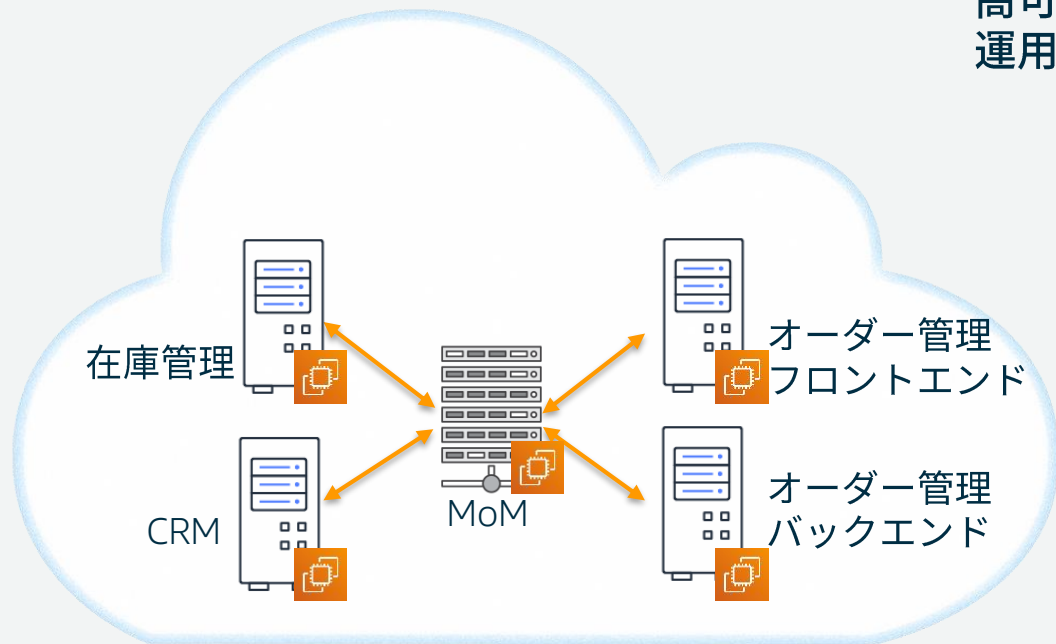
本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- その他



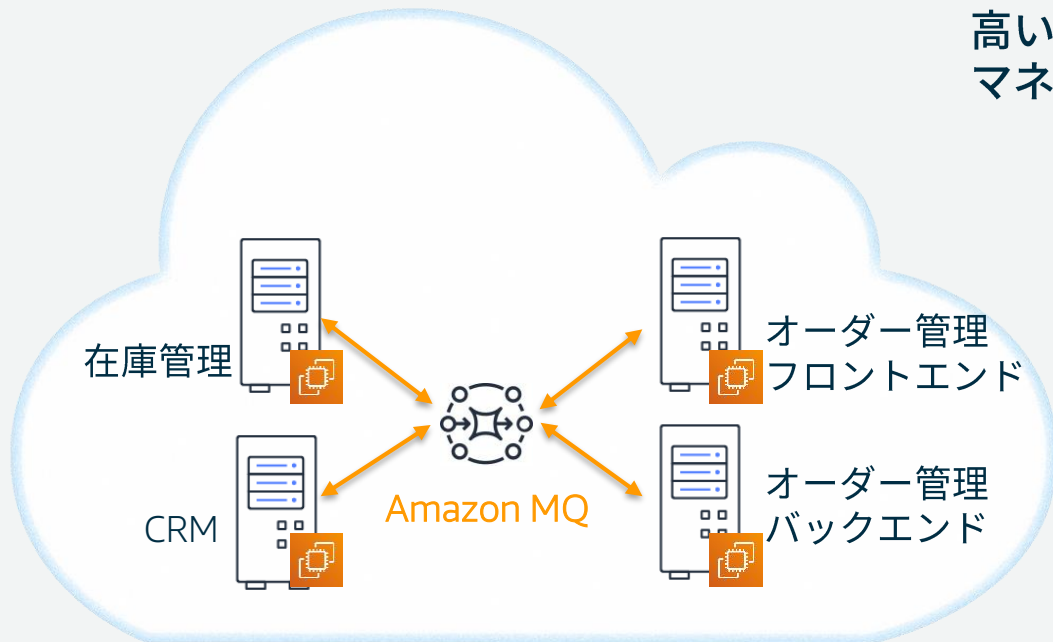
Amazon MQ のユースケース – オーバーヘッドと高額なライセンスの削減

ライセンス料は依然として高価
高可用性が難しい
運用スタッフが引き続き必要



Amazon MQ のユースケース – オーバーヘッドと高額なライセンスの削減

ライセンスが不要
高い可用性
マネージドサービス



移行における変更例 (JMS/Tomcat利用時, IBM MQとの比較)

Java (変更無し)

```
InitialContext initCtx = new InitialContext();
Context envContext = (Context) initCtx.lookup("java:comp/env");
ConnectionFactory connectionFactory = (ConnectionFactory) envContext.lookup("jms/ConnectionFactory");
Connection connection = connectionFactory.createConnection();
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
MessageProducer producer = session.createProducer((Destination) envContext.lookup("jms/queue/dev-queue-1"));
TextMessage producerMessage = session.createTextMessage("Test Message");
producer.send(producerMessage);
```

context.xml における JNDI 定義 (IBM MQ への接続)

```
<Resource name="jms/ConnectionFactory"
type="com.ibm.mq.jms.MQQueueConnectionFactory"
factory="com.ibm.mq.jms.MQQueueConnectionFactoryFactory"
auth="Container"
CHAN="DEV.APP.SVRCONN" HOST="*****" PORT="1414"
OMGR="QM1" TRAN="1"/>

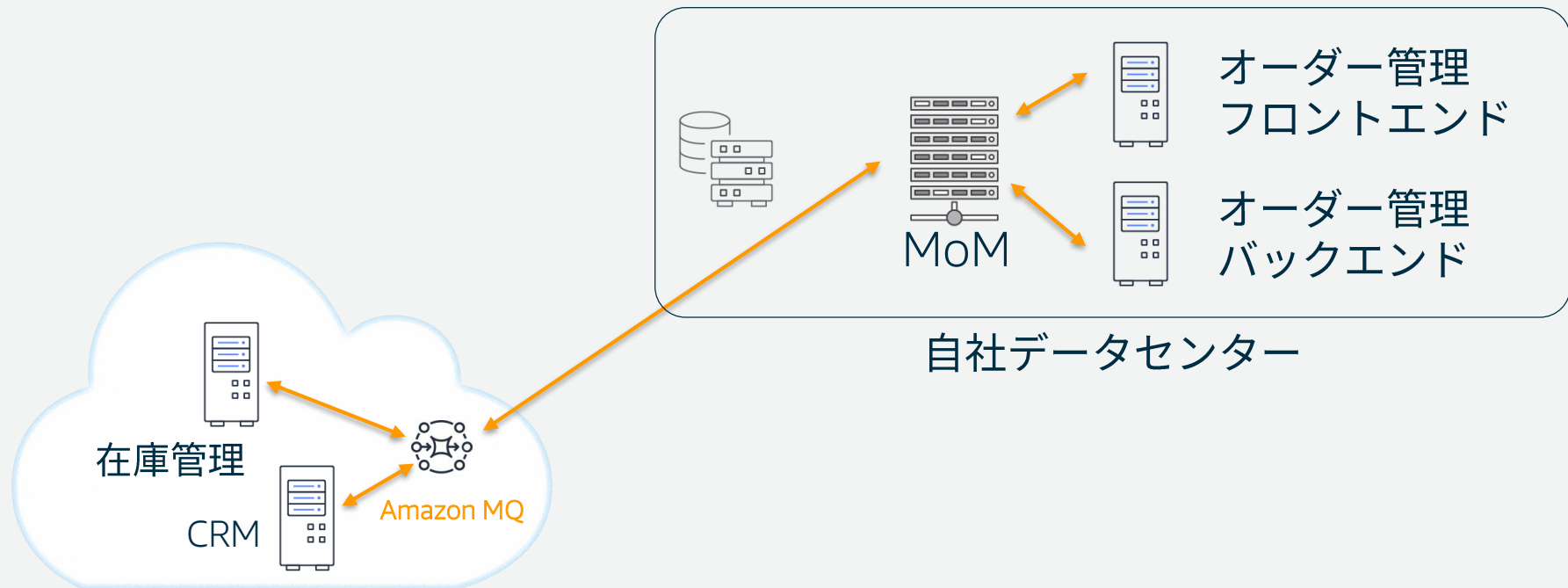
<Resource name="jms/queue/dev-queue-1"
type="com.ibm.mq.jms.MQQueue"
factory="com.ibm.mq.jms.MQQueueFactory"
auth="Container"
QU="DEV.QUEUE.1"/>
```

context.xml における JNDI 定義 (Amazon MQ への接続)

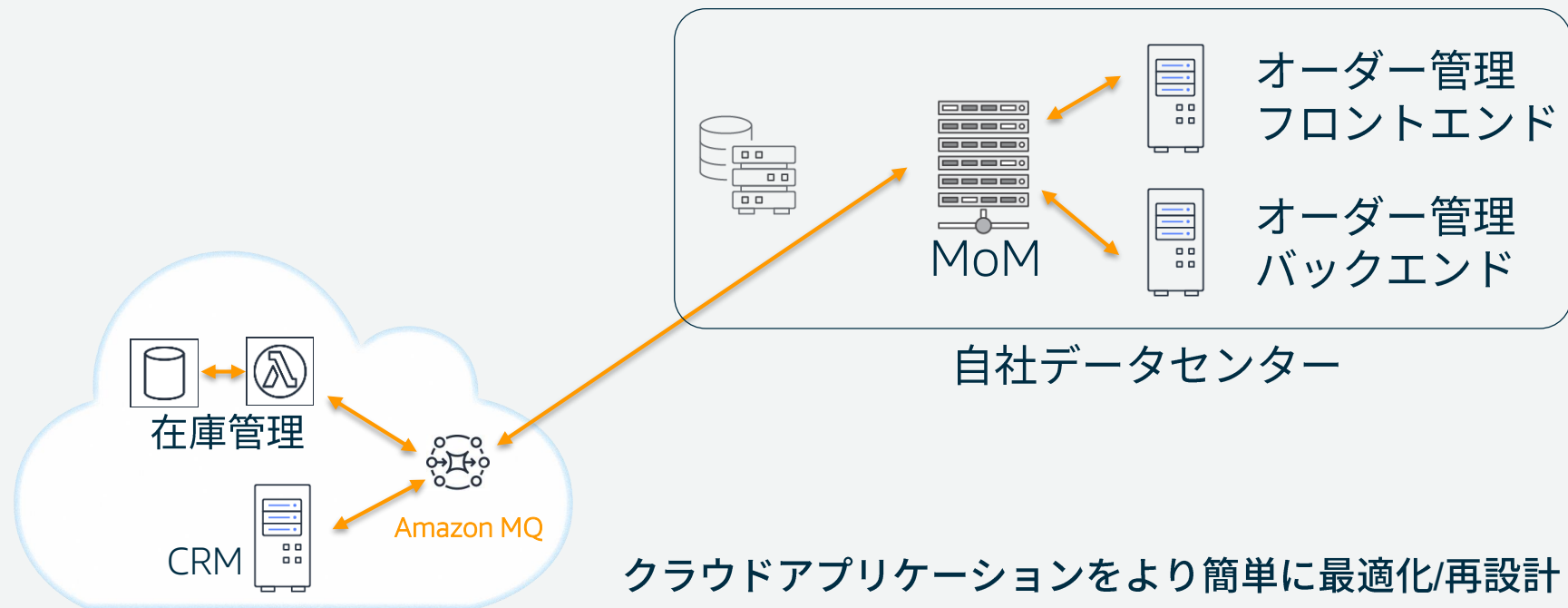
```
<Resource name="jms/ConnectionFactory"
type="org.apache.activemq.ActiveMQConnectionFactory"
factory="org.apache.activemq.jndi.JNDIReferenceFactory"
auth="Container"
brokerURL="ssl://*****.us-west-2.amazonaws.com:61617"
userName="*****" password="*****"
useEmbeddedBroker="false"/>

<Resource name="jms/queue/dev-queue-1"
type="org.apache.activemq.command.ActiveMQQueue"
factory="org.apache.activemq.jndi.JNDIReferenceFactory"
auth="Container"
physicalName="DEV.QUEUE.1"/>
```

Amazon MQ のユースケース – 高速な移行とリファクタリング



Amazon MQ のユースケース – 高速な移行とリファクタリング



Amazon MQ の適用例

大規模 製造事業者様:

これまでオンプレミスで商用のメッセージ指向ミドルウェアを利用されており、\$500Kのライセンス更新を迎えるタイミングでオンプレミスからAWSへの lift-and-shifted を行いました。コストとオーバーヘッド削減のためマネージドサービスへの移行を希望されていましたが、すべてのアプリケーションを書き換えるだけのリソースを確保することは困難でした。

中規模 出版事業者様:

クラウドへの移行開始に伴い、最初の3つのアプリケーションをクラウドへ移動しようと試みました。すべてのアプリケーションはオンプレミスの商用ブローカー経由で接続されていたため、オンプレミスとクラウド上のアプリケーションをブリッジするためには、標準インターフェースで既存のブローカーと接続可能なメッセージブローカーがクラウド上に必要でした。

グローバルなIT組織を持つ Fortune 500 企業様:

自ら管理、運用するメッセージング基盤を所有しており、アプリケーションチームは標準化された API (JMS) によってこの基盤を使用していました。既存のシステムをマネージドサービスへ置き換えることで運用のオーバーヘッドを削減しつつ、アプリケーションの書き直しを避けるためにアプリケーションへのインターフェースの互換性を保つ必要がありました。

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- その他



ベストプラクティス – セキュリティ

- パブリックアクセスビリティのないブローカーを選択する
Public IP がアサインされず、攻撃に晒されるリスクが低減する
- TLS の補完としてクライアント側の暗号化を使用する
AWS Encryption SDK 等によってアプリケーションレベルの暗号化を適用する
- 認可マップを設定する
authorizationEntry によってグループ単位でのアクセス制御を設定する
- システムグループを設定する
ActiveMQ Web コンソールから操作するリソースには activemq-webconsole グループを割り当て
- VPC セキュリティグループによって不要なプロトコルをブロックする

Amazon MQ のベストプラクティス:

https://docs.aws.amazon.com/ja_jp/amazon-mq/latest/developer-guide/amazon-mq-best-practices.html

ベストプラクティス – 接続性

- Amazon MQ Elastic Network Interface を変更または削除しない
ENI は Amazon MQ サービスによって管理される。誤って削除した場合、ブローカーの再作成が必要
- 接続プールを使用する
`org.apache.activemq.jms.pool.PooledConnectionFactory` クラスを使用することで Producer からの複数リクエストに対して Connection を再利用できる
- フェイルオーバートランスポートを使用する
- メッセージセレクトクを使用しない
Consumer にルーティングを行わせる事で、一般的に Producer との結合度があがる
- 永続サブスクリプションよりも仮想送信先を選択する
JMS の仕様によって永続サブスクリプションに接続できる Consumer は1つに限定される。

Amazon MQ のベストプラクティス:

https://docs.aws.amazon.com/ja_jp/amazon-mq/latest/developer-guide/amazon-mq-best-practices.html

ベストプラクティス – パフォーマンス

- コンシューマーが低速な場合 `concurrentStoreAndDispatchQueues` 属性を `false` にする
`true` の場合、File System への書き込みが最適化されず、都度シーケンシャルに行われる
- 最良なスループットのために正しいブローカーインスタンスタイプを選択する
以下のケースではインスタンスタイプの拡大でスループットを向上しやすい
 - 非永続メッセージを利用する場合
 - コンシューマーが高速である場合
 - バッチトランザクションを利用する場合
- ブローカーのネットワークを正しく設定する
 - 永続メッセージを利用する
 - ブローカーの Advisory Message を無効化しない
 - マルチキャストブローカーディスカバリーを使用しない
- Prepared 状態の XA トランザクションを復旧して低速な再起動を回避する

Amazon MQ のベストプラクティス:

https://docs.aws.amazon.com/ja_jp/amazon-mq/latest/developer-guide/amazon-mq-best-practices.html

本日のアジェンダ

- メッセージング概要
- Amazon MQ 概要
- Amazon MQ の構成要素
- Amazon MQ のユースケース
- ベストプラクティス
- その他



リージョン

Amazon MQ は東京リージョンを含む 14 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

コンプライアンス

- PCI DSS 準拠
- ISO 9001、27001、27017、27018 の認証を取得
- SOC 1、2、3 に準拠
- HIPAA に適合

2018年12月5日 Amazon MQ が PCI および ISO コンプライアンスプログラムをサポート
<https://aws.amazon.com/about-aws/whats-new/2018/12/amazon-mq-introduces-support-for-pci-and-iso-compliance-programs/>

2018年11月20日 Amazon MQ で SOC コンプライアンスプログラムのサポートを開始
<https://aws.amazon.com/about-aws/whats-new/2018/11/amazon-mq-introduces-support-for-soc-compliance-program/>

2018年10月2日 Amazon MQ が HIPAA 対応になりました
<https://aws.amazon.com/about-aws/whats-new/2018/10/amazon-mq-is-now-hipaa-eligible/>

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**
- 接続数(ワイヤレベルプロトコルあたり): **1,000**
 - ✓ mq.t2.micro ブローカーの場合: 100
- セキュリティグループ数: **5** (ブローカーあたり)
- CloudWatchでモニタリングされる送信先 (キューとトピック): **200**
 - ✓ 最初の200件の送信先のみがモニタリングの対象となります

設定

- Amazon MQ 設定数: **1,000**
- 設定あたりのリビジョン: **300**

ユーザー

- ブローカーあたりのユーザー数: **250**
- ユーザーあたりのグループ: **20**

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**
 - ✓ 新しい 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**
 - ✓ ActiveMQ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、選択するブローカー (単一ブローカーまたはアクティブ/スタンバイ ブローカー) による
- mq.m5.large のアクティブ/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
- 時間単位の使用 GB 数を加算した値を、月の使用時間数で割って算出
- **GB-Month あたり 0.36USD**

データ転送料金

- 同じリージョンの可用性ゾーンでブローカー間で転送されるトラフィックの場合、各方向に 0.01 USD/GBが課金
- クロスリージョンネットワークの場合、EC2 の料金体系に従って、各リージョン外の通信が課金対象
- 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 ブローカーを 1 か月あたり最大 750 時間、ストレージを最大 1 GB ご利用いただけます

よくあるご質問 - Amazon MQ と Amazon SQS/Amazon SNS 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

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/>
- Jeff Barr's Blog post: <https://aws.amazon.com/blogs/aws/amazon-mq-managed-message-broker-service-for-activemq/>
- Amazon MQ Forum:
 - <https://forums.aws.amazon.com/forum.jspa?forumID=279>
- Amazon MQ Twitch: <https://www.twitch.tv/videos/259845421>

7 月の Black Belt Online Seminar 配信予定

<https://amzn.to/JPWebinar>

~~7/3 (水) 18:00-19:00 Amazon MQ~~

7/5 (金) 18:00-19:00 AWS Summit TOKYO/OSAKA 2019 振り返り 2019 年主要アップデートまとめ

7/16 (火) 12:00-13:00 Amazon Personalize

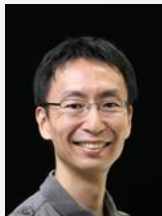
7/17 (水) 18:00-19:00 Amazon Simple Queue Service(SQS)

7/23 (火) 12:00-13:00 AWS CloudHSM

7/24 (水) 18:00-19:00 AWS Command Line Interface

7/30 (火) 12:00-13:00 Amazon CloudFront

7/31 (水) 18:00-19:00 Amazon ECS Deep Dive



AWS の日本語資料の場所「AWS 資料」で検索



The screenshot shows the AWS Japan homepage. At the top, there's a navigation bar with the AWS logo, links for '日本担当チームへお問い合わせ' (Contact our Japan team), 'サポート' (Support), '日本語' (Japanese), 'アカウント' (Account), and a 'コンソールにサインイン' (Sign in to the console) button. Below this is a secondary navigation bar with links for '製品' (Products), 'ソリューション' (Solutions), '料金' (Pricing), 'ドキュメント' (Documentation), '学習' (Learning), 'パートナー' (Partners), 'AWS Marketplace', 'その他' (Other), and a search icon. The main content area features a large heading 'AWS クラウドサービス活用資料集トップ' (AWS Cloud Service Usage Resource Collection Top). Below the heading is a paragraph in Japanese explaining that AWS is a secure cloud service platform supporting business scale and growth, and that the page provides Japanese resources and video content. At the bottom of the main content area, there are four buttons: 'AWS Webinar お申込' (AWS Webinar Registration), 'AWS 初心者向け' (For AWS Beginners), '業種・ソリューション別資料' (Resources by Industry and Solution), and 'サービス別資料' (Resources by Service).

aws

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]



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>

