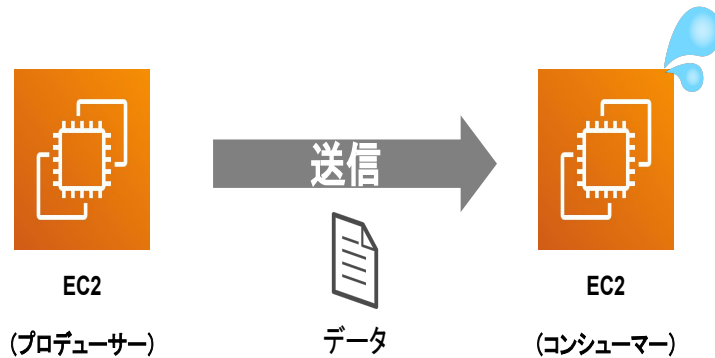




SQSについて説明します。

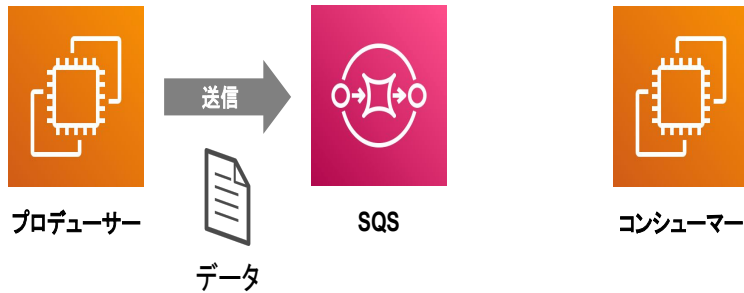
疎結合(デカップリング)でない場合の例



- SQSの理解の前に疎結合というキーワードを理解しましょう。デカップリングとも呼ばれます。
- まずは疎結合設計を採用していない、ダメな例を見ていきましょう
 - 例えば、ネットショッピングでお客さんからの注文情報を処理する場合、
 - EC2インスタンス内のアプリケーションで処理した注文の情報を、別のEC2インスタンスへ直接渡すような設計を考えてみます。
 - 疎結合の文脈においては、情報の発信元をプロデューサー、受信する側をコンシューマーと呼びます。
 - この図の場合、受け取り側のEC2インスタンスの処理負荷が高く、データを受け取れる状態になかった場合、処理は失敗します。
 - 受信側の状態を気にする必要が

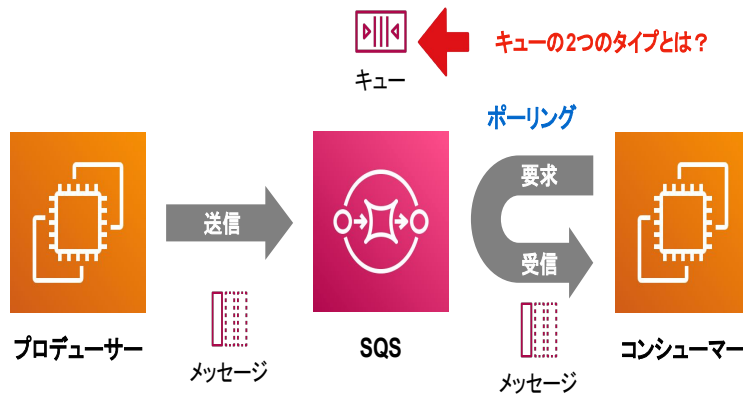
- あります。
- 処理に失敗したら重要な情報が失われて、ビジネス的な損失につながりかねません。単一障害点となるわけです。

SQSを使用した疎結合設計(デカップリング)



- では疎結合設計のパターンを考えます。
 - 2台のEC2で処理するのは同様ですが、
 - 間にSQSサービスを導入します
 - この方式を取ることで、送信側のアプリケーションはSQSへの送信のみを意識すればよく、受信側のアプリケーションの状態を気にする必要がなくなります。
 - SQSを使用すると、密接なアプリケーション同士が切り離されるため、実行や失敗は個別に行われ、システムの全体的な耐障害性が向上します。
 -

SQSは「メッセージキューイングサービス」



- なお、受け渡すデータのことを SQSではメッセージと呼び、メッセージを溜め込む場所をキューと呼びます。
 - このように、SQSはメッセージをキューに溜め込む「メッセージキューイング」サービスです。
 - 「メッセージキューイング」とは、システム間で送信するデータを一時的に溜め込む場所を設けて、そこから順次処理していく仕組みです。
 - 1台目のEC2は情報をSQSへ送信します
 - 2台目のEC2はSQSの情報が必要です。どのように受け取るでしょうか？ SQSが送りつけるのではなく...
 - ポーリングにより受け取ります。
 - ポーリングとはメッセージが溜まっていないかを定期的に確認しにいき、メッセージがあれば取得するという確認方式です。
 - 細かくは、キューの中のメッセー

- ジを削除するプロセスもありますが、一旦割愛します。
- SQSが直接EC2に情報を送りつけているわけではないことに注意してください。
 - コンシューマーのEC2が「処理する準備ができたから、情報を受け取るわ」という形で、コンシューマーEC2の都合によって受け取れます。
- これにより、もしも2台目のEC2が処理が忙しくなったとしても、適切に処理されます
- メッセージを溜め込む場所である「キュー」ですが、2つのタイプがあります。



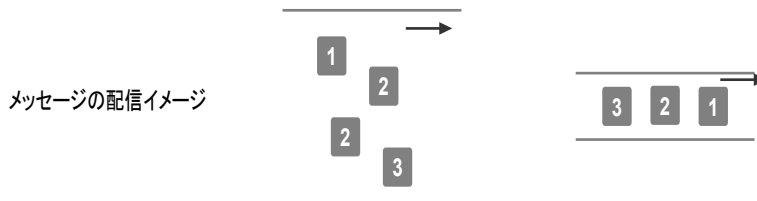
SQS

標準キュー

FIFOキュー

SQS スタンダード(標準)キューとFIFOキュー

	スタンダードキュー	FIFOキュー
メッセージの配信回数	1回以上	1回のみ
メッセージの配信順序	順序が変わる可能性あり	順序を保証する
スループット	ほぼ無制限	最大300件／1秒



- SQSキューの2つのタイプ、スタンダードキュー(標準キュー)とFIFOキューです。
 - まずはスタンダードキューです。
 - これは受信したメッセージの順序が保証されない。
 - 公式ドキュメントはこれを「ベストエフォート型の順序を提供」と記載があります。
 - 場合によっては同じメッセージを複数回受信する場合がある。
 - イメージ図をみてください。メッセージが1,2,3、もうひとつ2がありますね。
 - ということで、アプリケーション側で制御するか、または次のFIFOキューを使うべきです。
 - FIFOキューはFirst In First Outの略です(先入先出)
 - 先に入れたものを先に出す順序、これは例え

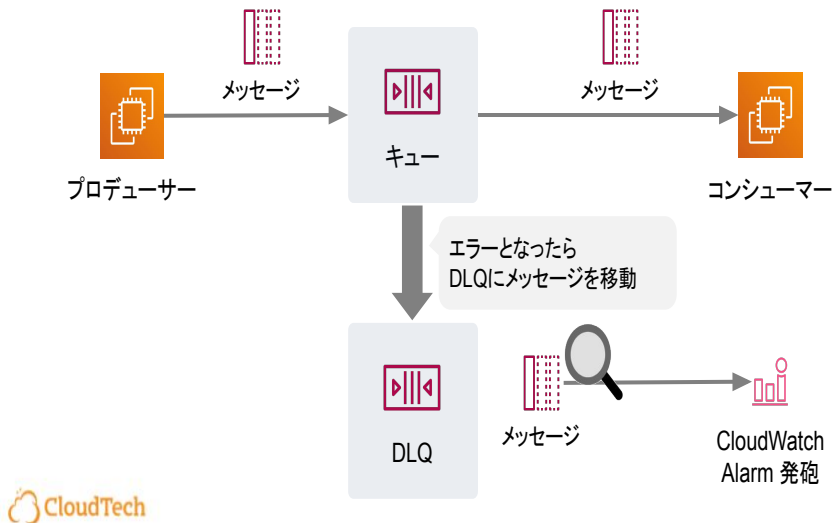
- ば、コンビニで売っている紙パックの牛乳、お店に入荷して、賞味期限が近いものから順番に売っていきたいですね。
 - 先に入れたものから出していきましょう、これが先入先出です。
- 受信したメッセージの順序を正確に保ち、配信となります
- 標準キューは複数回の可能性があったが、FIFOキューの場合は必ず1回のみの配信を保証
- スループットに関しては、FIFOが最大1秒あたり300件のメッセージを処理できる性能です。
 - なお、現在、高スループットモードというのもあり、10倍の3,000件の処理性能があります。
- アプリの作りが順序をクリティカルな場合、アプリ側のロジックで対応するかFIFOを使うべきですね。
 - それぞれの特徴は試験にもよく出てきますので、FIFOの方がより信頼できるということを覚えておいてください。



デッドレターキュー
遅延キュー
ショートポーリング
ロングポーリング

SQS デッドレターキュー

- デッドレターキュー (DLQ)



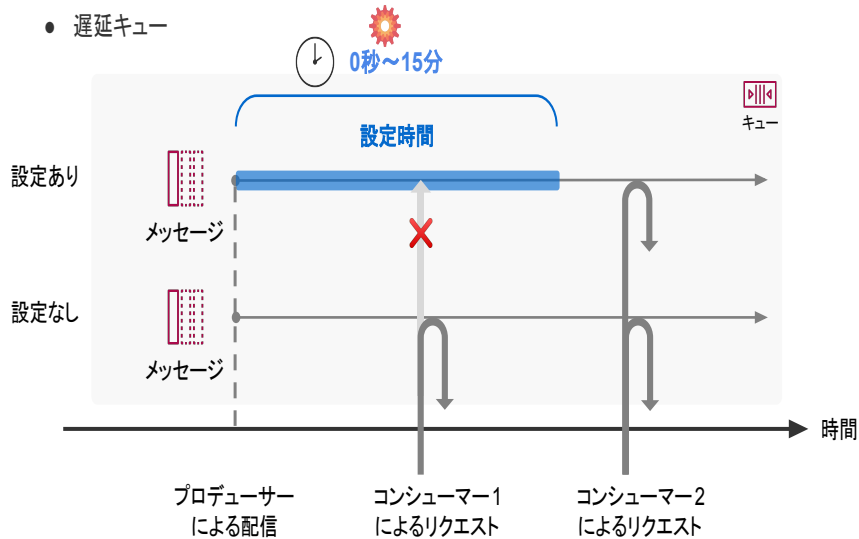
- デッドレターキューについて説明します。これはエラー調査に使われるもので、先ほどの標準や FIFOと同列のタイプではありません。
 - キューの送信が失敗したケースを考えます
 - たとえば、SQSの最大受信数に到達した場合や、後続処理のロジック変更などにより、キューの連携が失敗した場合
 - エラーとなりメッセージがデッドレターキューに送信されます。
 - キューというのは、メッセージを溜め込む場所のことでしたね。
 - その入れ物が、作成した標準キューとかFIFOキューとかとは別の場所、DLQという入れ物に移動される
 - そして、そのキューの中身を確認し、失敗した原因を

■ 調べる、という流れです。

- ここで大切なのは、デッドレターキューとは標準や FIFO と同列のタイプではなく、標準キューや FIFO キューで失敗したメッセージを貯めておく場所ということです。
 - またデッドレターキューにメッセージが入ったタイミングで CloudWatch と連携して通知を飛ばすことも可能なため、障害の早期発見に役立てることが出来ます。

SQS 遅延キューとメッセージタイマー

- 遅延キュー

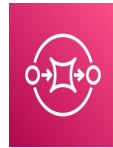


- 次に、遅延キューについて
 - アプリケーションによっては、メッセージ送信後、一定時間経過してからコンシューマーに処理をさせたいケースもあるかと思います。
 - その場合は、遅延キュー、もしくはメッセージタイマーを検討しましょう。
 - 上が設定しているもの
 - 下が通常の遅延キュー設定ないもの
- この機能はメッセージがキューに入ってから、指定時間が経過するまで、コンシューマーはそのメッセージを取得することができなくなります。
 - 通常のキューは取得できますが、遅延キューを設定したものは一定時間、取得できません。
 - 一定時間経過後、取得できるようになります。
 - 0秒~15分の間で設定可能

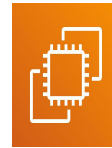
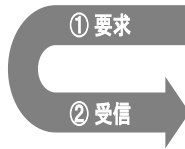
- なお、遅延キューはキューに入ったメッセージすべてに適用する場合に設定し、メッセージ個別に適用したい場合はメッセージタイマーを設定します。こちらはスタンダードキューの機能となり、FIFOキューではサポートされていません。

SQS ショートポーリング、ロングポーリング

- ショートポーリング



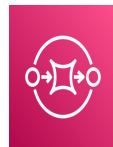
SQS



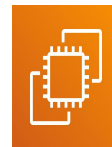
コンシューマー



- ロングポーリング



SQS



コンシューマー



- SQSのポーリング方式には「ショートポーリング」と「ロングポーリング」があります。
- ショートポーリングは、コンシューマーのメッセージ取得要求に対して、即応答をする方式です。
キューにメッセージが存在する場合はメッセージを返却し、存在しない場合は空を返却します。
- 一方ロングポーリングは、最大 20秒応答を待機します。
キューにメッセージが存在する場合は即メッセージを返却しますが、
存在しない場合は一定時間待機し、その間にメッセージが取得できればそのタイミングでメッセージを返却します。
一定時間待機してもメッセージが取得できない場合は、空のデータを返却します。

それではどちらの方式を採用すればいいのでしょうか？
利用シーンに寄りますが、多くのケースではロングポーリングを

採用します。

ショートポーリングは即応答となるため、APIコールの回数が増えて利用料金が増加するためです。



問題

アプリケーションを作成中で、アプリケーションコンポーネントを互いに切り離すためにメッセージバスが必要です。

このアプリケーションはバッチ処理を行わずに、1秒間に最大300のメッセージを生成します。メッセージが一度だけ配信され、キューに重複が発生しないことを保証する必要があります。メッセージの順序を維持する必要はありません。

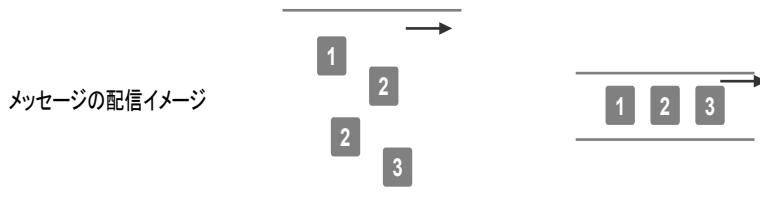
どのSQSキュー・タイプを使用すべきですか？

1. ロングポーリングキュー
2. 標準的なキュー
3. FIFOキュー
4. LIFOキュー



SQS スタンダード(標準)キューとFIFOキュー

	スタンダードキュー	FIFOキュー
メッセージの配信回数	1回以上	1回のみ
メッセージの配信順序	順序が変わる可能性あり	順序を保証する
スループット	ほぼ無制限	最大300件／1秒



設問の要件は、重複したメッセージをキューに入れることができないということです。

この理由があることから、FIFO キューを使用しなければなりません。

「メッセージの順序を維持する必要はありません。」という記述は、標準キューが使用できると思われそうですが、キューが重複しないことを保証するものではありません。

FIFO (first-in-first-out) キューは、メッセージ送受信を正確な順序で一度だけ処理します。

FIFO キューは、各メッセージが一度だけ配送され、コンシューマーがそれを処理し、削除するまで利用可能であることを意味します。

"標準的なキュー" は不正解です。
標準キューは各メッセージが少なくとも一度は配送されることを

意味する at-least-once 配送となります。

"ロングポーリングキュー" は不正解です。ロングポーリングはキューに適用できる設定であり、キューの種類ではありません。

"LIFOキュー" は存在しないため、不正解です。

参考文献

<https://aws.amazon.com/sqs/features/>