

# Microsoft SQL Server を AWS にデプロイするための ベストプラクティス

2020 年 5 月



## 注意

お客様は、この文書に記載されている情報を独自に評価する責任を負うものとします。このドキュメントは、(a) 情報提供のみを目的としており、(b) AWS の現行製品とプラクティスを表したものであり、予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤー、またはライセンサーからの契約義務や確約を意味するものではありません。AWS の製品やサービスは、明示または暗示を問わず、いかなる保証、表明、条件を伴うことなく「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で行われるいかなる契約の一部でもなく、そのような契約の内容を変更するものでもありません。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 目次

はじめに .....	1
高可用性と災害対策.....	2
アベイラビリティゾーンとマルチ AZ 配置.....	3
クラスタープレイスメントグループと拡張ネットワーキング .....	5
マルチリージョン配置.....	7
災害復旧.....	9
パフォーマンスの最適化.....	11
Amazon Elastic Block Store (Amazon EBS) の使用.....	12
インスタンスストレージ.....	15
Amazon FSx for Windows ファイルサーバー.....	16
帯域幅とレイテンシー .....	17
リードレプリカ.....	18
セキュリティの最適化.....	18
Amazon VPC.....	19
保管時の暗号化.....	19
転送時の暗号化.....	20
使用中の暗号化.....	20
AWS Key Management Service (AWS KMS).....	21
セキュリティパッチ .....	21
コスト最適化.....	21

Amazon EC2 CPU 最適化 .....	22
SQL Server Standard Edition への切り替え.....	23
z1d EC2 インスタンスタイプ .....	24
アクティブなレプリカライセンスの排除.....	25
運用上の優秀性.....	28
可観測性と根本原因の分析.....	28
平均修復時間の短縮 (MTTR).....	29
パッチ管理.....	29
まとめ .....	30
寄稿者 .....	30
ドキュメントの改訂.....	30

## 要約

このホワイトペーパーは、AWS で Microsoft SQL Server を実行する際に最低限のコストで価値を最大化するためのベストプラクティスにフォーカスしています。多くの汎用ユースケースにおいて Amazon Relational Database Service (Amazon RDS) for Microsoft SQL Server は簡単かつ迅速なソリューションを提供しますが、このペーパーでは特殊な要件を満たすために限界を押し広げる必要のあるシナリオにフォーカスしています。

特に、このホワイトペーパーでは、コストを最小限に抑え、SQL Server データベースの可用性を最大化し、パフォーマンスを最大化するためにインフラストラクチャを最適化し、セキュリティコンプライアンスを強化しながら、継続的なメンテナンスのために運用上の優秀性を実現する方法について説明します。AWS のサービスの柔軟性と Microsoft SQL Server の機能を組み合わせることで、アプリケーションを最適化し、ビジネスを変革する革新的なアプローチを求める人のために、拡張された機能を提供できます。

このホワイトペーパーの主な焦点は、Microsoft SQL Server 2019 で利用できる機能です。これは公開時点の最新バージョンです。以前のバージョン (2008、2012、2014、2016、2017) で実行されている既存のデータベースは、Microsoft SQL Server 2019 に移行し、互換モードで実行できます。

Microsoft SQL Server 2000、2005、2008 のメインストリームサポートおよび延長サポートは、[Microsoft によって終了](#)されました。これらのバージョンの Microsoft SQL Server で実行されているデータベースは、まず最初にサポートされているバージョンにアップグレードする必要があります。これらのバージョンの Microsoft SQL Server を AWS で実行することは可能ですが、このホワイトペーパーでは取り上げません。

## はじめに

AWS では、Microsoft SQL Server にベストマッチするクラウドを提供しています。現在も将来においても、Windows ベースのアプリケーションを実行するのに最適なクラウドプラットフォームです。Windows または Linux 上の Amazon EC2 で Microsoft SQL Server を使用すれば、数分以内にキャパシティを増減できます。数日はおろか数時間もかかりません。1 つのサーバーインスタンスでも、数百、数千のインスタンスでも、同時に作動させられます。

AWS には、10 年以上にわたって Windows ワークロードを実行してきた実績があります。AWS は、業界第 2 位のクラウドプロバイダーのほぼ 2 倍 ([IDC レポート調査](#)) の Windows Server インスタンスを実行しています。AWS は、Microsoft SQL Server のデプロイと実行に引き続き最も推奨されるオプションです。これは、AWS が提供する幅広いサービスと機能のユニークな組み合わせにより、Microsoft SQL Server ワークロードに最適なプラットフォームを提供することによるものです。

Microsoft SQL Server を実行するための要件は、多くの場合、次のカテゴリに分類されます。

- 高可用性と災害対策
- パフォーマンス
- セキュリティ
- コスト
- モニタリングとメンテナンス

これらの要件は、[AWS Well-Architected フレームワーク](#)の 5 つの柱に直接マッピングされます。

- 信頼性
- パフォーマンス効率
- セキュリティ

- コスト最適化
- 運用上の優秀性

このホワイトペーパーでは、これらの各要件について、さらに詳しく説明するとともに、AWS のサービスを使用したベストプラクティスについて説明します。

## 高可用性と災害対策

すべてのビジネスは、運用上の要件に対応できるデータソリューションを探しています。これらの要件は、多くの場合、目標復旧時間 (RTO) および目標復旧時点 (RPO) の特定の値に変換されます。RTO は、ビジネスがデータベースとアプリケーションの停止に耐えられる期間を示し、RPO によって許容されるデータ損失の量が決まります。例えば、1 時間の RTO は、アプリケーションが停止した場合の不運なイベントの場合、復旧計画は 1 時間以内にアプリケーションをオンラインに戻すことを目指す必要があることを示しています。同様に、RPO がゼロの場合、アプリケーションに影響する軽微な問題や重大な問題があってもアプリケーションがオンラインに戻った後にデータが失われないことを示します。

RTO 要件と RPO 要件の組み合わせにより、採用すべきソリューションが決まります。通常、RPO および RTO 値がゼロに近いアプリケーションでは、高可用性 (HA) ソリューションを使用する必要がありますが、災害対策 (DR) ソリューションは価値の高いアプリケーションに使用できます。多くの場合、HA ソリューションと DR ソリューションを組み合わせると、より複雑な要件に対応することもできます。

Microsoft SQL Server には、複数の高可用性と災害復旧 (HA/DR) のソリューションがあり、それぞれ特定の要件に適しています。次の表は、これらのソリューションを比較的に示しています。

表 1: Microsoft SQL Server の HA/DR オプション

ソリューション	HA	DR	Enterprise Edition	Standard Edition
ログ配布	いいえ	はい	はい	はい

ミラーリング (廃止)	はい	はい	はい	はい (完全な安全性のみ)
Always On 可用性グループ (AG)	はい	はい	はい	はい (2 個のノード) <sup>1</sup>
Always On フェイルオーバークラスターインスタンス	はい	いいえ	はい	はい (2 個のノード)
分散可用性グループ	はい	はい	はい	いいえ

これらのソリューションは、Microsoft SQL Server をアクティブスタンバイまたはパッシブスタンバイとして実行している 1 つ以上のセカンダリサーバーに依存します。特定の HA/DR 要件に基づいて、これらのサーバーは相互に近い場所に配置することも、遠くに配置することもできます。

AWS では、レイテンシーを低くするか、失敗の可能性を極めて低くするかを選択できます。これらのオプションを組み合わせ、ユースケースに最適なソリューションを作成することもできます。このホワイトペーパーでは、これらのオプションと、それらを Microsoft SQL Server ワークロードで使用方法について説明します。

## アベイラビリティゾーンとマルチ AZ 配置

AWS アベイラビリティゾーンは、別々の障害ドメインを提供するように設計されており、ワークロードを低レイテンシーで通信するために比較的近接するように設計されています。アベイラビリティゾーンは、ミラーリング、Always On 可用性グループ、基本可用性グループ、またはフェイルオーバークラスターインスタンスを使用したデータベースの同期レプリケーションに適したソ

<sup>1</sup>Microsoft SQL Server 2019 Standard Edition の [Always On 基本可用性グループ](#)は、可用性グループごとの単一のデータベースに対して最大 2 つのパッシブレプリカ（プライマリレプリカに加えて）をサポートします。HA モードで複数のデータベースが必要な場合は、データベースごとに別個の可用性グループを定義する必要があります。Microsoft SQL Server 2016 および 2017 ではレプリカの最大数は 1 となります。基本可用性グループは、Microsoft SQL Server の以前のバージョンでは使用できません。

リューションです。Microsoft SQL Server はデータ損失をゼロにし、アベイラビリティゾーンの低レイテンシーインフラストラクチャと組み合わせると、高いパフォーマンスを実現します。

これは、ほとんどのオンプレミスのデプロイと AWS の主な違いの 1 つです。例えば、Always On フェイルオーバークラスターインスタンス (FCI) は、単一のデータセンター内で使用されることがよくあります。これは、FCI クラスターのすべてのノードが同じ共有ストレージにアクセスできる必要があるためです。異なるデータセンターにこれらのノードを配置すると、パフォーマンスが低下する可能性があります。ただし、AWS では、FCI ノードを別々のアベイラビリティゾーンに配置できますが、リージョン内のすべてのアベイラビリティゾーン間のネットワークリンクが低レイテンシーであるため、高いパフォーマンスを実現できます。

この機能により、より高いレベルの可用性が実現され、3 番目のノードが不要になります。このノードは、多くの場合、災害復旧の目的で FCI クラスターと結合されます。

Microsoft SQL Server FCI は、FCI に参加するすべてのノードからアクセス可能な共有ストレージに依存しています。[Amazon FSx for Windows ファイルサーバー](#)は、2 つのアベイラビリティゾーン間で自動的にデータをレプリケートし、自動障害検出、フェイルオーバー、およびフェイルバックによる高可用性を実現する共有ストレージを提供するフルマネージド型サービスです。また、SMB 継続的可用性 (CA) 機能を完全にサポートします。これにより、Microsoft SQL Server Always On のデプロイを簡素化し、MSSQL FCI のストレージ層として Amazon FSx を使用できます。

パフォーマンスのチューニングとコストの最適化に Amazon FSx が適用されるシナリオについては、このドキュメントの以降のセクションで説明します。

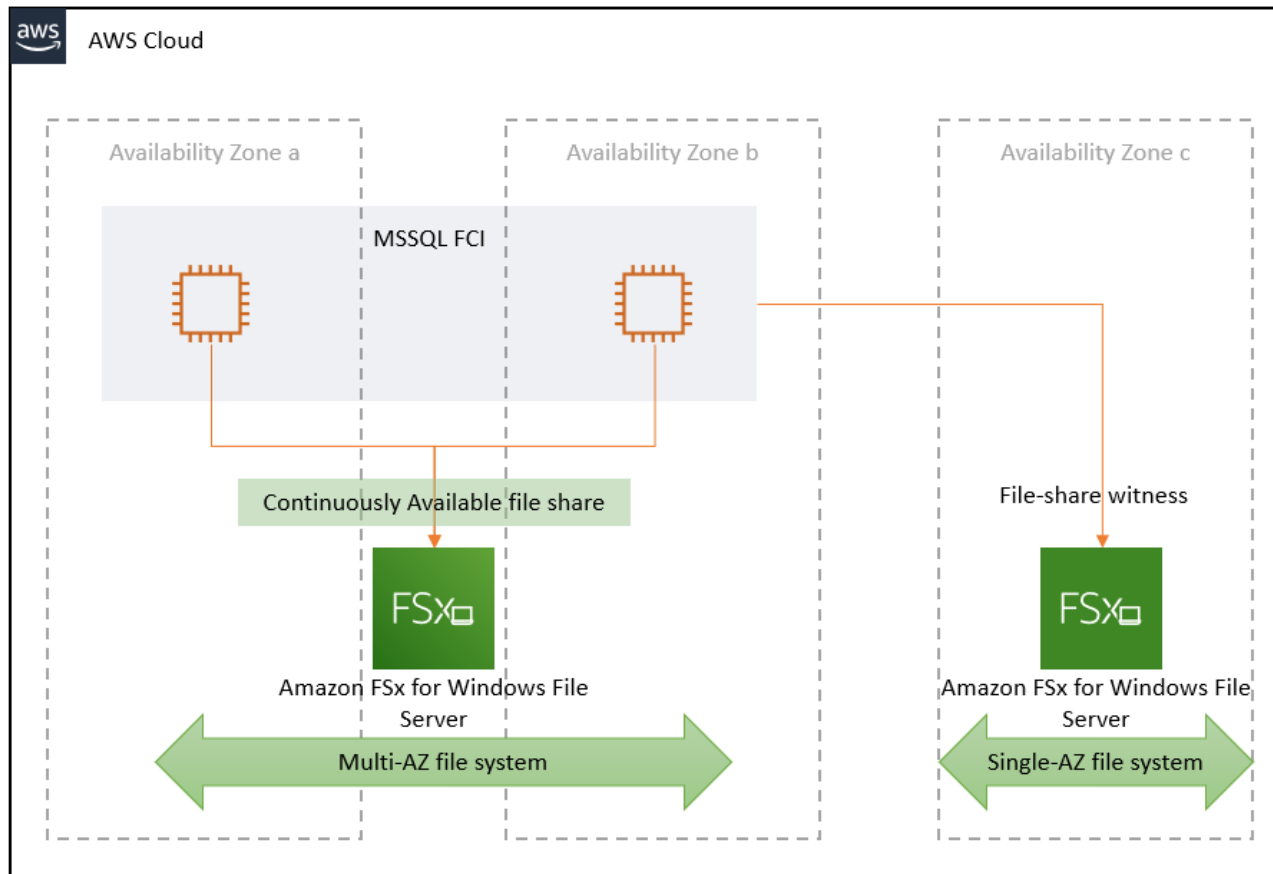


図 1: AWS の Microsoft SQL Server Always On フェイルオーバークラスターインスタンス

## クラスタープレイズメントグループと拡張ネットワーキング

従来、可用性が高くなるとパフォーマンスが低下します。AWS では、この二分法の差異を軽減する方法があります。

Amazon EC2 では、クラスタープレイズメントグループ内に多数の EC2 インスタンスをデプロイできます。つまり、これらの EC2 インスタンスは単一の AZ 内にあるだけでなく、ネットワークレイテンシーを最小限にするため、同じデータセンター内で物理的に近い場所に配置できます。

AWS で最大の帯域幅を得るには、拡張ネットワーキングと [Elastic Network Adapter \(ENA\)](#)、ま

たは新しい [Elastic Fabric Adapter \(EFA\)](#) を活用できます。このアダプタは、m5n、m5dn、r5n、r5dn の各インスタンスと組み合わせると、最大 100 Gbps の帯域幅を提供できます。

一見すると、サーバーに近接すると同時に障害が発生する可能性が高くなるため、HA 要件と競合しているように見えることがあります。しかしながら、このアプローチはマルチ AZ と組み合わせ、より高いレベルの可用性を提供できます。例えば、クラスタープレイスメントグループ内に Always On FCI を作成し、非同期レプリケーションを使用してそのクラスターをマルチ AZ Always On 可用性グループにノードとして追加できます。この配置では、プロセスまたはインスタンスが失敗した場合に、別のローカルインスタンスへの自動フェイルオーバーを提供しながら、パフォーマンスを最大限に高めることができます。

さらに、AZ に障害が発生した場合は、データベースを手動で 2 番目の AZ にフェイルオーバーできます。アベイラビリティゾーン間のレイテンシーは比較的 low、非同期レプリケーションが実施されていても、データ損失はほぼゼロです。これは、同じ AWS リージョン内の DR ソリューションと組み合わせ、レイテンシーの影響を受けるアプリケーションのパフォーマンス低下がほぼゼロの HA の例です。

前述のとおり、クラスタープレイスメントグループを使用すると、ネットワークレイテンシーは最小限に抑えられますが、クラスタープレイスメントグループ内で実行されているインスタンスで同時に障害が発生する可能性が高くなります。例えば、2 つの EC2 インスタンスがあり、両方が同じ物理ホストで実行されている場合、または両方が同じ電源とネットワークスイッチを共有している場合、これらの基盤となるコンポーネントのいずれかに障害が発生すると、両方の EC2 インスタンスで障害が発生します。代わりに、EC2 インスタンスで同時に障害が発生する可能性を低下させる代わりに、クラスタープレイスメントグループのレイテンシーを低く抑えることができます。このような場合、スプレッドプレイスメントグループを活用して、インスタンスを十分に離して配置し、同時に部分的または完全な障害の可能性を最小限に抑えることができます。

## マルチリージョン配置

予定外のイベントに対してさらに高い回復力が必要なワークロードについては、AWS のグローバル規模を活用して、ほぼすべての状況下で可用性を確保できます。

デフォルトでは、Amazon Virtual Private Cloud (Amazon VPC) は単一のリージョン内に限定されているため、マルチリージョンのデプロイでは、異なるリージョンにデプロイされている Microsoft SQL Server インスタンス間の接続を確立する必要があります。AWS では、これを行う方法は数多くあり、それぞれがさまざまな要件に適しています。

1. [VPC ピアリング](#): 2 つの VPC 間で暗号化されたネットワーク接続を提供します。トラフィックは AWS ネットワーキングバックボーンを通過するため、レイテンシーやその他のインターネットの危険を排除します。
2. [AWS Transit Gateway](#): 複数の VPC またはオンプレミスサイトを接続する必要がある場合は、AWS Transit Gateway を使用して、VPC 間のネットワーク接続を確立するための管理と設定のオーバーヘッドを簡素化できます。
3. [VPN 接続](#): AWS VPN ソリューションは、ハイブリッド環境で運用して、AWS VPC をオンプレミスのサイトやクライアントに接続する必要がある場合に特に便利です。
4. [VPC 共有](#): アプリケーションまたは他のクライアントが複数の AWS アカウントに分散している場合、Microsoft SQL Server インスタンスをすべてのアカウントで使用できるようにする簡単な方法は、VPC 共有を使用することです。共有 VPC は、AWS Transit Gateway、AWS VPN CloudHub、VPN 接続、または VPC ピアリング接続を使用して他の VPC に接続することもできます。これらの接続は、ワークロードが複数のアカウントとリージョンに分散されている場合に便利です。

Microsoft SQL Server インスタンスに接続する必要があるリモートリージョンにデプロイされているアプリケーションまたはユーザーがある場合は、すべての Direct Connect 接続からすべての AWS リージョンへの接続を提供する [AWS Direct Connect](#) 機能を使用できます。

複数リージョンの Microsoft SQL Server デプロイで同期レプリケーションを行うことは可能ですが、選択したリージョンが離れているほど、同期レプリケーションのパフォーマンスが低下します。マルチリージョンデプロイのベストプラクティスは、特に地理的に離れたリージョンに対して非同期レプリケーションを確立することです。積極的な RPO 要件を伴うワークロードの場合、非同期マルチリージョンデプロイをマルチ AZ またはシングル AZ 同期レプリケーションと組み合わせることができます。3 つの方法すべてを 1 つのソリューションに組み合わせることもできます。ただし、これらの組み合わせにより Microsoft SQL Server ライセンスコストが大幅に増加します。これは、計画の一部として考慮する必要があります。

複数のリージョンにまたがる複数のレプリカが含まれる場合、[分散可用性グループ](#)が最も適している可能性があります。この機能により、各リージョンにデプロイされた可用性グループを、より大きな分散可用性グループに組み合わせることができます。

分散可用性グループを使用して、リードレプリカの数を増やすこともできます。従来の可用性グループでは、最大 8 個のリードレプリカを使用できます。つまり、プライマリを含む合計 9 個のレプリカを持つことができます。分散可用性グループを使用すると、最初の可用性グループに 2 番目の可用性グループを追加し、レプリカの合計数を 18 に増やすことができます。このプロセスは、3 番目の可用性グループと 2 番目の分散可用性グループで繰り返すことができます。2 番目の分散可用性グループは、1 番目または 2 番目の可用性グループをプライマリとして含めるように設定できます。分散可用性グループは、Microsoft SQL Server Always On がほぼ無制限のスケールを実現できる手段です。

分散可用性グループのもう 1 つのユースケースは、ダウンタイムなしのデータベース移行です。移行中に移行対象先で読み取り専用レプリカを使用できます。Microsoft SQL Server 分散可用性グループが Active Directory と Windows Server フェイルオーバークラスター (WSFC) から独立しているのは、これらのケースの主なメリットです。これにより、Active Directory や WSFC の複雑さを心配することなく、移行の両側を同期させることができます。詳細については、こちらの[ブログ記事](#)を参照してください。

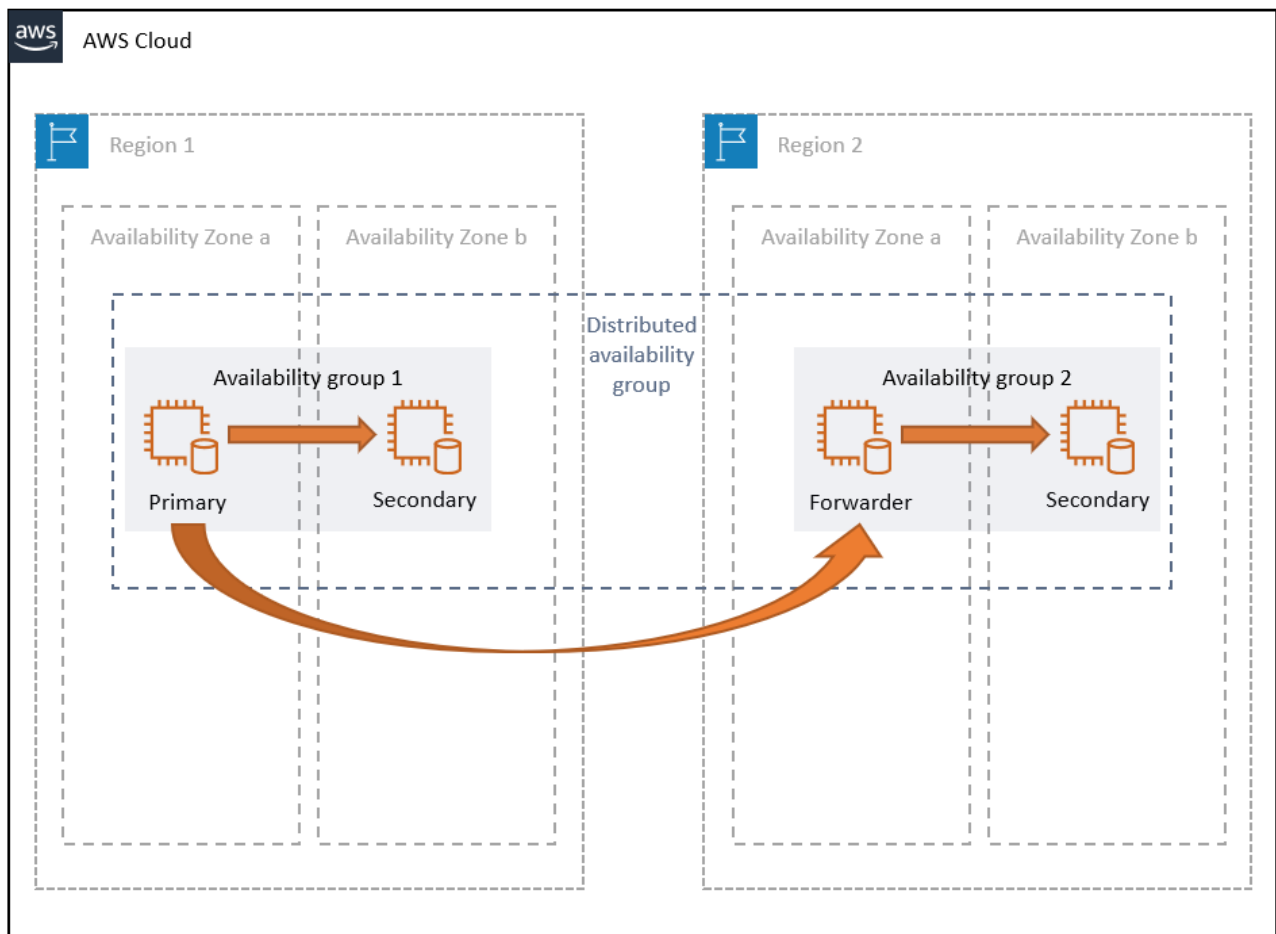


図 2: AWS での Microsoft SQL Server 分散可用性グループ

## 災害復旧

高可用性 (HA) ソリューションと同様に、災害対策 (DR) ソリューションでは、別のサーバーの Microsoft SQL Server データベースのレプリカも必要です。しかしながら、DR の場合は、他のサーバーは多くの場合、プライマリサイトから遠く離れたリモートサイトにあります。つまり、同期レプリケーションを使用する HA ソリューションに依存している場合、レイテンシーが高くなり、パフォーマンスが低下します。

DR ソリューションは、データの非同期レプリケーションに依存することがよくあります。HA と同様に、DR ソリューションはブロックレベルまたはデータベースレベルのレプリケーションにも基

づいています。例えば、Microsoft SQL Server のログ SHIPPING はデータベースレベルでデータをレプリケートしますが、Windows 記憶域レプリカはブロックレベルのレプリケーションを実装するために使用できます。

DR ソリューションは、コスト、RPO、RTO、複雑さ、各ソリューションを実装するための取り組みなどの要件に基づいて選択されます。

AWS では、ログ SHIPPING や Windows 記憶域レプリカなどの一般的な Microsoft SQL Server DR ソリューションに加えて、[CloudEndure Disaster Recovery](#) も提供しています。

CloudEndure Disaster Recovery を使用すると、ダウンタイムを数分に短縮し、1 秒未満の RPO でデータ損失を防ぎ、実装を簡素化し、信頼性を高め、総所有コストを削減できます。

CloudEndure は、オペレーティングシステム、インストールされているすべてのアプリケーション、およびすべてのデータベースを含む仮想マシン全体をステージング領域にレプリケートするエージェントベースのソリューションです。ステージング領域には、CloudEndure Disaster Recovery によって自動的にプロビジョニングと管理が行われる、低コストのリソースが含まれています。これにより、重複するリソースのプロビジョニングにかかる費用を大幅に削減できます。ステージング領域で実際のワークロードが実行されている訳ではないので、重複するソフトウェアライセンスやハイパフォーマンスコンピューティングに費用はかかりません。むしろ、低コストのコンピューティングやストレージに対して料金が発生します。復旧したワークロードに必要なとされる、適切なサイズのコンピューティングや高性能なストレージを備えた、完全にプロビジョニングされた災害対策環境のみが、災害や演習のときに起動します。

また、AWS では、移行プロジェクトに追加コストなしで CloudEndure を利用できます。

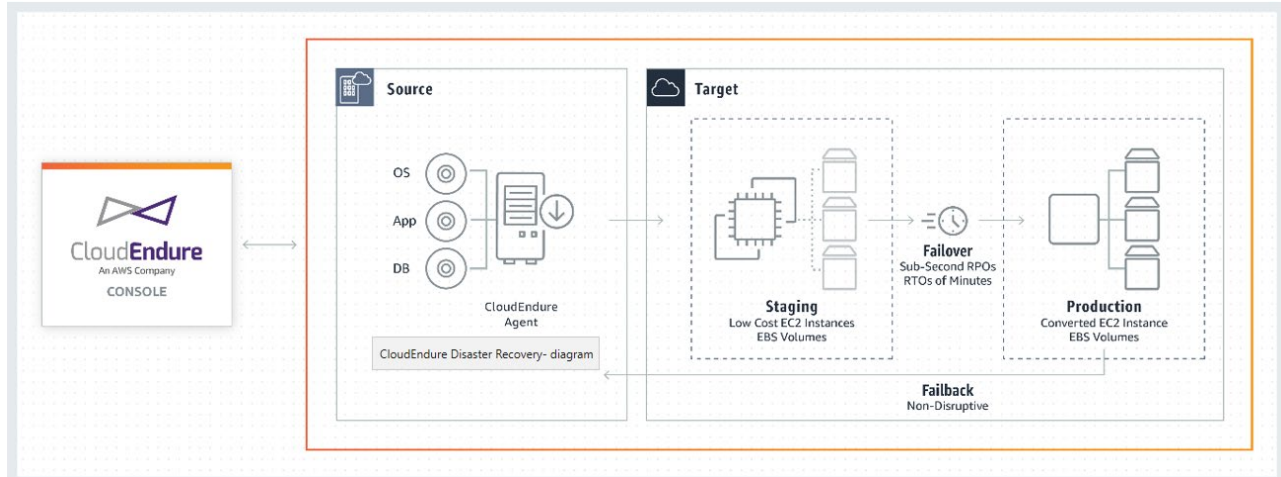


図 3: CloudEndure Disaster Recovery

## パフォーマンスの最適化

場合によっては、パフォーマンスの最大化が最優先事項になることがあります。Microsoft SQL Server と AWS のどちらにも、ワークロードのパフォーマンスを大幅に向上させるためのいくつかのオプションがあります。

ワークロードのパフォーマンスを向上させる最初の最も効果的な方法は、アプリケーションとデータベーススキーマを最適化することです。また、従来のリレーショナルデータベースではなく専用のデータベースソリューションを使用するようにアプリケーションを変更することで、アプリケーションのパフォーマンスを大幅に向上させることもできます。アプリケーションの最新化のための AWS データベースオプションには、キー値およびドキュメントストレージ用の [Amazon DynamoDB](#)、グラフデータ用の [Amazon Neptune](#)、[Amazon Quantum Ledger Database](#)、[Amazon Managed Blockchain](#) などがあります。

リレーショナルデータベースを使用する必要があるアプリケーションを最新化するために、多くの企業が Amazon Aurora を選択しています。Amazon Aurora は、クラウド向けに構築されたオープンソースのエンタープライズグレードの RDBMS マネージド型サービスで、MySQL および PostgreSQL と完全に互換性があります。Amazon Aurora は、標準的な [MySQL](#) データベースと

比べて最大で 5 倍、標準的な PostgreSQL データベースと比べて最大で 3 倍高速です。また、商用データベースと同等のセキュリティ、可用性、信頼性を、10 分の 1 のコストで実現します。

Amazon Aurora は [Amazon Relational Database Service \(RDS\)](#) を使ったフルマネージド型サービスであるため、ハードウェアのプロビジョニング、データベースのセットアップ、パッチ適用、バックアップといった時間のかかる管理タスクが自動化されます。また、Aurora のサーバーレス機能により、サーバーを管理することなく、最高レベルのコスト最適化と安定したパフォーマンスを実現できます。

パフォーマンスチューニングで考慮すべきもう 1 つの領域は、物理データベーススキーマです。慎重に物理設計して、非クラスター化インデックスを定義し、テーブルとインデックスの水平パーティション化を使用することで、データベースのパフォーマンスを大幅に向上させることができます。トランザクションデータの分析処理には、Microsoft SQL Server の機能である列ストアインデックスを活用できます。このインデックスは、最大 10 倍のパフォーマンスとデータ圧縮の改善を提供できます。

場合によっては、データを論理的に小さなチャンクにスライスし、各チャンクを別々のサーバーにホストすることで、スケーリングの問題を解決できます。この手法はシャーディングと呼ばれ、書き込み量の多いデータベースのスケーリングに特に便利です。

ただし、すでにアプリケーションと DB レベルの最適化を行っている場合や、状況がそのようなプラクティスに役立たない場合があります (例えば、レガシー基幹業務アプリケーションの実行や、差し迫った負荷増加の予測など)。このような場合、次はインフラストラクチャに重点を置き、それをどのように再設定して問題を解決または軽減できるかを評価します。次のセクションでは、AWS クラウドで Microsoft SQL Server のパフォーマンスを最大化するためのインフラストラクチャ設計オプションに焦点を当てます。

## Amazon Elastic Block Store (Amazon EBS) の使用

Amazon EBS は Single-AZ ブロックストレージサービスで、様々な要件に対応するためのオプションを多数備えています。1 つのボリュームで一貫した予測可能な結果でパフォーマンスを最大化

する場合、[プロビジョンド IOPS](#) ボリュームタイプ (io1) を使用するのが最も簡単です。io1 EBS ボリュームあたり最大 64,000 IOPS (16 KiB I/O サイズに基づく) と 1000 MB/ 秒のスループットをプロビジョニングできます。

単一の EBS ボリュームよりも高い IOPS とスループットが必要な場合は、複数のボリュームを作成して Windows または Linux インスタンスでストライプ化できます (Microsoft SQL Server 2017 以降は Windows と Linux の両方のシステムにインストールできます)。ストライプ化により、インスタンスあたりの利用可能な IOPS を最大 80,000、インスタンスあたりのスループットは最大 2,375 MB/ 秒まで増やすことができます。

覚えておく必要がある点の 1 つは、EBS 最適化 EC2 インスタンスタイプを使用することです。つまり、EC2 インスタンスとそれにアタッチされた EBS ボリューム間のリクエストを処理するために、専用ネットワーク接続が割り当てられます。

プロビジョンド IOPS (io1) ボリュームを 1 つ使用して IOPS とスループットの要件を満たすことはできますが、汎用 SSD (gp2) ボリュームは、適切に設定すると SQL Server ワークロードの価格とパフォーマンスのバランスが良くなります。

汎用 SSD (gp2) ボリュームでは、レイテンシーは 1 桁台のミリ秒であり、長時間で 3,000 IOPS までバーストする機能を備えています。この機能は SQL Server に適しています。SQL Server などのリレーショナルデータベースにより生じる IOPS 負荷は、頻繁にスパイクする傾向があります。例えば、テーブルスキャンオペレーションではスループットのバーストが必要ですが、他のトランザクションオペレーションでは一貫した低レイテンシーが必要です。

EBS ボリュームを使用する主な利点の 1 つは、ポイントインタイムで瞬間的な EBS スナップショットを作成できることです。この機能は、EBS スナップショットを Amazon S3 インフラストラクチャにコピーします。このインフラストラクチャは 99.999999999% の耐久性を提供します。

EBS ボリュームが単一の AZ に設定されていても、EBS スナップショットは同じリージョン内の任意の AZ に復元できます。ブロックレベルのスナップショットはデータベースバックアップと同じではなく、データベースバックアップのすべての機能がこの方法で実現できるわけではありません。

ん。そのため、この方法は多くの場合、通常のデータベースバックアップ計画と組み合わせて補完されます。

各 EBS ボリュームのサイズは最大 16 TB ですが、すべてのデータを Amazon S3 に転送するには時間がかかることがあります。EBS スナップショットは常にポイントインタイムです。つまり、バックグラウンドでデータが転送されている間は、SQL Server や他のアプリケーションが EBS ボリュームとの間で読み書きを続けることができます。

スナップショットからボリュームを復元すると、アプリケーションは読み取りおよび書き込みオペレーションにすぐに使用できるようになります。ただし、パフォーマンス容量が最大になるまでに時間がかかります。[Amazon EBS 高速スナップショット復元](#)を使用すると、ブロックに初めてアクセスするときの I/O 操作のレイテンシーを排除できます。高速スナップショット復元を使用して作成されたボリュームでは、プロビジョンドパフォーマンスをすべて即座に提供できます。

AWS Systems Manager Run Command を使用すると、オンラインの SQL Server ファイルの[アプリケーションコンシステントな EBS スナップショット](#)をいつでも取得できます。データベースをオフラインにしたり、読み取り専用モードにしたりする必要はありません。スナップショットプロセスでは、Windows Volume Shadow Copy Service (VSS) を使用して、VSS 対応アプリケーションのイメージレベルバックアップを取得します。Microsoft SQL Server は VSS 対応であり、この手法と完全に互換性があります。Linux インスタンスの VSS スナップショットを作成することもできますが、Linux は VSS をネイティブにサポートしていないため、このプロセスには手動ステップが必要です。

また、オーケストレーターアプリケーションを使用せずに、Windows または Linux EC2 インスタンスにアタッチされた複数の EBS ボリュームにわたって[クラッシュ整合性スナップショット](#)を作成することもできます。この方法を使用すると、コミットされないトランザクションとディスクにフラッシュされない書き込みのみが失われます。SQL Server は、クラッシュ時間前の整合性のあるポイントにデータベースを復元できます。

EBS ボリュームはシンプルで使いやすく、ほとんどの場合効果的です。ただし、Amazon EBS を使用して達成できるよりも高い IOPS とスループットが必要な場合があります。

# インスタンスストレージ

[ストレージ最適化 EC2 インスタンスタイプ](#)は、固定サイズのローカルディスクを使用し、様々なストレージテクノロジーを使用できます。その中で、Non-Volatile Memory Express (NVMe) は、最高の IOPS とスループットを備えた最速のテクノロジーです。インスタンスタイプの i3 クラスには、NVMe SSD ドライブ (例: i3.16xlarge) があり、それぞれに 1.9 TB のストレージがある 8 つのディスクが付属しています。パフォーマンスを最大化するためにストレージ最適化 EC2 インスタンスタイプを選択する場合、一部の小さいインスタンスタイプは、他のインスタンスと共有されるインスタンスストレージを提供することを理解することが重要です。これらは、物理ホストにアタッチされた物理ディスクに存在する仮想ディスクです。i3.2xlarge などのより大きなインスタンスタイプを選択することで、インスタンスストアディスクと基盤となる物理ディスクの間に 1 対 1 の対応があることを確認できます。これにより、一貫したディスクパフォーマンスが確保され、ノイズの多いネイバー問題が解消されます。

インスタンスディスクは一時的なものであり、関連付けられた EC2 インスタンスが存在する間のみ存続します。EC2 インスタンスに障害が発生した場合、または停止または終了した場合、そのインスタンスストレージディスクはすべて消去され、インスタンスに保存されたデータは回復できません。EBS ボリュームとは異なり、インスタンスストレージディスクはスナップショットを使用してバックアップできません。そのため、永続データに EC2 インスタンスストレージを使用する場合は、耐久性を高める方法を提供する必要があります。

インスタンスストレージに適した用途の 1 つは、[tempdb](#) システムデータベースファイルです。これらのファイルは Microsoft SQL Server サービスが再起動されるたびに再作成されるためです。SQL Server は、シャットダウン中にすべての tempdb 一時テーブルとストアードプロシージャを削除します。ベストプラクティスとして、tempdb ファイルはユーザーデータベースとは別に、高速ボリュームに保存する必要があります。最高のパフォーマンスを得るには、同じファイルグループ内の tempdb データファイルが同じサイズで、ストライピングされたボリュームに保存されていることを確認します。

EC2 インスタンスストレージのもう 1 つの用途は、[バッファプールの拡張](#)です。バッファプールの拡張は、Microsoft SQL Server の Enterprise Edition と Standard Edition の両方で使用できます。この機能では、RAM と永続的ディスクストレージとの間のセカンダリキャッシュとして高速ランダムアクセスディスク (SSD) が使用されるため、SQL Server でワークロードを実行するときのコストとパフォーマンスのバランスが取れます。

インスタンスストレージディスクは EC2 インスタンスで使用できる最速ですが、そのパフォーマンスは物理ディスクの速度に上限があります。複数のディスクにまたがってストライピングすることで、1 つのディスクの最大値を超えることができます。

また、記憶域スペース (単一の Windows インスタンスの場合) および記憶域スペースダイレクト (Windows Server フェイルオーバークラスターの場合) のストレージプールのキャッシュレイヤーとして、インスタンスストレージディスクを使用することもできます。

## Amazon FSx for Windows ファイルサーバー

[Amazon FSx for Windows ファイルサーバー](#)は、Amazon EC2 での Microsoft SQL Server のもう 1 つのストレージオプションです。このオプションは、3 つの主要なユースケースに適しています。

- フェイルオーバークラスターインスタンスに加わる SQL Server ノードによって使用される共有ストレージとして
- Windows Server フェイルオーバークラスター上の任意の SQL Server クラスターで使用するファイル共有監視として
- 専用 [EBS 最適化](#)で利用できるよりも高いスループットレベルを達成するオプションとして

最初の 2 つのケースについては、このドキュメントの前のセクションで説明しました。2 番目のケースをよりよく理解するために、EBS スループットは EC2 インスタンスのサイズに依存することに注意してください。EC2 インスタンスサイズが小さいほど EBS スループットが低くなるため、EBS のスループットを高めるには、より大きなインスタンスサイズが必要です。ただし、インスタ

ンスサイズが大きいほど、コストが高くなります。ワークロードがネットワーク帯域幅の大部分を未使用のままにしても、基盤となるストレージにアクセスするためにより高いスループットが必要な場合、SMB 経由の共有ファイルシステムを使用すると、より小さい EC2 インスタンスサイズを使用することでコストを削減しながら、必要なパフォーマンスを引き出すことができます。

Amazon FSx は、ファイルシステムあたり最大 2 GB/ 秒、数十万の IOPS、安定したミリ秒未満のレイテンシーで、高速な[パフォーマンス](#)を提供します。ワークロードに適したパフォーマンスを実現できるよう、スループットレベルをファイルシステムのサイズとは別個に選択できます。スループットキャパシティのレベルが高くなると、ファイルサーバーがアクセスする SQL Server インスタンスに対して処理できる IOPS レベルも高くなります。ストレージ容量によって、保存できるデータの量だけでなく、ストレージで実行できる I/O オペレーション / 秒 (IOPS) の数も決まります。ストレージの GB ごとに 3 つの IOPS が提供されます。各ファイルシステムは、最大 64 TB のサイズにプロビジョニングできます。

## 帯域幅とレイテンシー

パフォーマンスを調整するときは、レイテンシーと帯域幅の違いを覚えておくことが重要です。ネットワークのレイテンシーと可用性のバランスを取る必要があります。ENA ドライバーを使用すると帯域幅が最大になりますが、レイテンシーには影響しません。ネットワークレイテンシーは、相互接続ノード間の距離と直接相関して変化します。クラスタリングノードは可用性を高める 1 つの方法ですが、クラスターノードを互いに近づきすぎると、同時障害の可能性が高くなり、可用性が低下します。これらを離しすぎると、可用性が最も高くなりますが、レイテンシーは高くなります。

各 AWS リージョン内の AWS アベイラビリティーゾーンは、ほとんどの実用的なケースに適合するように設計されています。各 AZ は、他のアベイラビリティーゾーンから物理的に分離されるように設計されており、地理的に近接してネットワークレイテンシーを低く抑えます。そのため、大多数の事例で、クラスターノードを複数のアベイラビリティーゾーンに分散させることがベストプラクティスになります。

## リードレプリカ

DB トランザクションの多くが読み取り専用クエリであり、多数の着信接続がデータベースをあふれさせていると判断される場合があります。リードレプリカは、この状況に対する既知のソリューションです。プライマリの SQL Server インスタンスから 1 つ以上のリードレプリカインスタンスに読み取り専用トランザクションをオフロードできます。リードレプリカはバックアップオペレーションの実行にも使用でき、バックアップウィンドウ中のパフォーマンスヒットからプライマリインスタンスを解放します。可用性グループリスナーを使用する場合、接続文字列を読み取り専用としてマークすると、Microsoft SQL Server は受信接続をすべての使用可能なリードレプリカにルーティングし、読み取り / 書き込みトランザクションのみをプライマリインスタンスに送信します。

Microsoft SQL Server 2012 で導入された [Always On 可用性グループ](#) は、最大 4 つのセカンダリレプリカをサポートします。Microsoft SQL Server の最新バージョン (2014、2016、2017、2019) では、Always On 可用性グループは 1 セットのプライマリデータベースと 1~8 セットの対応するセカンダリデータベースをサポートします。

地理的に分散した場所からデータベースに接続するユーザーまたはアプリケーションがある場合があります。レイテンシーが懸念される場合は、ユーザーやアプリケーションに近いリードレプリカを見つけることができます。

読み取り専用トランザクションにセカンダリデータベースを使用する場合は、サーバーソフトウェアのライセンスが適切に付与されていることを確認する必要があります。

## セキュリティの最適化

クラウドセキュリティは AWS の最優先事項であり、多数の AWS セキュリティ機能を利用できます。これらの機能を Microsoft SQL Server の組み込みセキュリティ機能と組み合わせることで、最も厳格な要件や期待に応えることができます。

## Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) には、転送中のデータ保護に役立つ多くの機能があります。セキュリティグループを使用して EC2 インスタンスへのアクセスを制限し、特定のエンドポイントとプロトコルのみを許可できます。ネットワークアクセスコントロールリストを使用して、既知の脅威の原因を拒否することもできます。

ベストプラクティスは、SQL Server インスタンスを VPC 内のプライベートサブネットにデプロイし、VPC NAT ゲートウェイまたはカスタム NAT インスタンスを介したインターネットアクセスのみを許可することです。

## 保管時の暗号化

EBS ボリュームを使用して SQL Server データベースファイルを保存する場合は、ブロックレベルの暗号化を有効にするオプションがあります。Amazon EBS は、暗号化と復号を透過的に処理します。これはシンプルなチェックボックスから使用でき、それ以上のアクションは必要ありません。Amazon FSx for Windows ファイルサーバーには、保管時の暗号化機能が組み込まれています。Amazon EBS と Amazon FSx はいずれも、暗号化キーを管理するために AWS Key Management Service (AWS KMS) と統合されています。つまり、AWS KMS を介して、AWS が提供するキーを使用するか、独自のキーを使用することができます。詳細については、[AWS KMS のドキュメント](#)を参照してください。

データベースレベルでは、SQL Server Transparent Data Encryption ([TDE](#)) を使用できます。これは Microsoft SQL Server で利用できる機能で、保管時のデータの透過的な暗号化を提供します。TDE は Amazon RDS for SQL Server で利用でき、EC2 インスタンスの SQL Server ワークロードでも有効にできます。

以前は、TDE は SQL Server Enterprise Edition でのみ使用可能でした。ただし、SQL Server 2019 では Standard Edition でも利用可能になっています。以前のバージョンの SQL Server の

Standard Edition でデータベースファイルを保管時に暗号化する場合は、代わりに EBS 暗号化を使用できます。

EBS 暗号化と TDE のトレードオフの違いを理解することが重要です。EBS 暗号化はブロックレベルで行われます。つまり、データは保存時に暗号化され、取得時に復号されます。ただし、TDE では、暗号化はファイルレベルで行われます。つまり、データベースファイルは暗号化され、対応する証明書を使用してのみ復号できます。

例えば、TDE なしで EBS 暗号化を使用し、データベースデータまたはログファイルを EC2 インスタンスから暗号化が有効になっていない S3 バケットにコピーした場合、ファイルは暗号化されません。さらに、誰かが EC2 インスタンスにアクセスした場合、データベースファイルは即座に公開されます。

ただし、EBS 暗号化を使用するときにパフォーマンスが低下することはありませんが、TDE を有効にすると、サーバーリソースにさらなるオーバーヘッドが追加されます。

## 転送時の暗号化

ベストプラクティスとして、SSL/TLS プロトコルを使用して Microsoft SQL Server ワークロードの転送時の暗号化を有効にできます。Microsoft SQL Server は[暗号化された接続](#)をサポートしており、AWS の SQL Server ワークロードも例外ではありません。さらに、SQL Server ストレージレイヤーに SMB プロトコルを使用する場合、Amazon FSx は、SQL Server や他のアプリケーションの設定を変更せずに、ファイルシステムにアクセスするときに SMB 暗号化を使用して転送中のすべてのデータを[自動的に暗号化](#)します。

## 使用中の暗号化

Microsoft SQL Server では、クライアント証明書を使用して機密データを保護するために [Always Encrypted](#) が用意されているため、分離を提供することで、データを所有して表示できる人とデータを管理するがアクセス権を持たない人を分離できます。この機能は、Amazon RDS for SQL Server と Amazon EC2 の SQL Server ワークロードの両方でも利用できます。

## AWS Key Management Service (AWS KMS)

AWS KMS は、暗号化キーを作成して保存するためのフルマネージド型サービスです。AWS KMS で生成されたキーを使用するか、独自のキーを使用できます。いずれの場合も、キーが AWS KMS から送信されることはなく、不正なアクセスから保護されます。AWS KMS キーを使用して SQL Server バックアップファイルを Amazon S3、Amazon S3 Glacier、またはその他のストレージサービスに保存するときに暗号化できます。Amazon EBS 暗号化は AWS KMS と統合されています。

## セキュリティパッチ

一般的なセキュリティ要件の 1 つは、セキュリティパッチとアップデートの定期的なデプロイです。AWS では、[AWS Systems Manager Patch Manager](#) を使用してこのプロセスを自動化できます。Patch Manager のユースケースは、セキュリティパッチに限定されないことに注意してください。詳細については、このホワイトペーパーの「運用上の優秀性」トピックのパッチ管理に関する説明を参照してください。

## コスト最適化

SQL Server は、License Included (LI) や、Bring Your Own License (BYOL) ライセンスモデルを通して AWS でホストできます。LI では、AWS で SQL Server を実行し、AWS の時間単位の使用料金の一部として、ライセンスの料金を支払います。このモデルの利点は、長期契約を必要とせず、いつでも製品の使用を停止し、その使用量に対する支払いを停止できることです。

ただし、多くの企業が SQL Server ライセンスに多額の投資を行っているため、既存のライセンスを AWS で再利用したい場合があります。この場合は BYOL の使用が可能です。

- ソフトウェアアシュアランス (SA) をお持ちの場合は、その利点の 1 つに、[ソフトウェアアシュアランスによる Microsoft ライセンスモビリティプログラム](#)があります。このプログラムでは、Amazon EC2 インスタンスなど、どこでも実行されているサーバーインスタンスでライセンスを使用できます。
- SA をお持ちでない場合は、Amazon EC2 Dedicated Hosts を使用して AWS でご自分のライセンスを使用できる場合があります。詳細については、AWS での Microsoft ワークロードに関するよくある質問のライセンスセクションを参照して、ライセンスコンプライアンスを確認してください。

EC2 Dedicated Hosts の BYOL オプションを使用すると、EC2 ホストの物理コア数とそのホストで利用可能な vCPU の合計数の約半分になるため、コストを大幅に削減できます。ただし、このオプションの一般的な課題の 1 つに、ライセンスの使用状況とコンプライアンスを追跡する難しさがあります。[AWS License Manager](#) では、ライセンスの使用状況を追跡し、必要に応じてライセンスコンプライアンスを適用することで、この問題の解決を支援します。AWS License Manager は、追加料金なしで AWS のお客様が利用できます。

## Amazon EC2 CPU 最適化

z1d インスタンスタイプは最大の CPU 能力を提供するので、コンピューティング集約型の SQL Server デプロイで CPU コア数を減らすことができます。ただし、SQL Server のデプロイはコンピューティング集約型ではなく、メモリやストレージなどの他のリソースに強度を提供する EC2 インスタンスタイプが必要な場合があります。これらのリソースを提供する EC2 インスタンスタイプは、要件以上の固定数のコアも提供しているため、最終的に、まったく使用されないコアのライセンスコストが高くなる可能性があります。幸い、AWS はこのような状況に対応するソリューションを提供しています。[EC2 CPU 最適化](#)を使用すると、EC2 インスタンスで使用できるコア数を減らし、不要なライセンスコストを回避できます。

## SQL Server Standard Edition への切り替え

SQL Server のエンタープライズグレードの機能は、Enterprise Edition でのみ使用できます。ただし、時間の経過とともに Standard Edition でもこれらの機能の多くが使用可能になったため、Enterprise Edition をこれらの機能にのみ使用していた場合は、Standard Edition に切り替えることができます。1 つの例は、Transparent Data Encryption (TDE) を使用した保管時の暗号化です。これは、Microsoft SQL Server 2019 の時点で Standard Edition で利用可能になりました。

Enterprise Edition を使用する最も一般的な理由の 1 つは、常にミッションクリティカルな HA 機能です。しかしながら、可用性を低下させることなく Standard Edition に切り替えることができる代替オプションがあります。これらのオプションを使用して、SQL Server のデプロイのコストを最適化できます。

1 つのオプションは、[Always On 基本可用性グループ](#)を使用することです。このオプションは Always On 可用性グループと似ていますが、いくつかの制限があります。最も重要な制限は、基本の可用性グループに 1 つのデータベースのみを持つことができることです。この制限により、複数のデータベースに依存するアプリケーションではこのオプションが除外されます。

もう 1 つのオプションは、Always On フェイルオーバークラスターインスタンス (FCI) を使用することです。FCI はインスタンスレベルで HA を提供するため、SQL Server インスタンスでホストされているデータベースの数は関係ありません。従来、このオプションは単一のデータセンター内の HA に制限されていました。ただし、前述で説明したように、Amazon FSx for Windows ファイルサーバーを使用してこの制限を克服できます。このドキュメントの「高可用性と災害対策」のセクションを参照してください。

次のシナリオでは、Amazon FSx を使用して Microsoft SQL FCI デプロイを実行する複雑さとコストを簡素化できます。

- FCI 用の共有ストレージソリューションの実装には複雑さとコストがあるため、可用性グループと SQL Server Enterprise Edition の使用を選択する場合があります。しかしながら、Standard Edition に切り替えて、ライセンスコストを大幅に削減し、SQL デプロイと継続的な管理の全体的な複雑性を簡素化できるようになりました。
- サードパーティのストレージレプリケーションソフトウェアソリューションを使用して、共有ストレージで SQL Server FCI をすでに使用している場合があります。その場合は、ストレージレプリケーションソリューションのライセンスを購入し、共有ストレージソリューションを自分でデプロイ、管理、保守する必要があります。Amazon FSx でフルマネージド型の共有ストレージソリューションの使用に切り替えることができるので、SQL Server FCI デプロイのコストを簡素化および削減できます。
- Microsoft SQL Server Always On デプロイをオンプレミスで実行している場合、FCI と AG-FCI を組み合わせてプライマリデータセンターのサイト内で高可用性を実現し、AG はサイト間の災害復旧ソリューションを提供します。アベイラビリティゾーンの組み合わせと、複数のアベイラビリティゾーンにまたがってデプロイされた高可用性共有ストレージのための Amazon FSx のサポートにより、個別の HA と DR ソリューションが不要になり、コストを削減し、デプロイの複雑さを簡素化できるようになりました。

Amazon FSx を使用した Microsoft SQL Server FCI デプロイの詳細については、ブログ記事 [Simplify your Microsoft SQL Server high availability deployments using Amazon FSx for Windows File Server](#) を参照してください。

## z1d EC2 インスタンスタイプ

ハイパフォーマンスの z1d インスタンスは、Microsoft SQL Server や Oracle データベースなど、ライセンスコストが高いワークロードに最適化されています。z1d インスタンスは、カスタムのインテル Xeon スケーラブルプロセッサを使用して構築されており、最大 4.0 GHz の持続的な全コアターボ周波数を実現します。これは他のインスタンスよりも大幅に高速です。より高速なシケンシャル処理を必要とするワークロードの場合、z1d インスタンスで実行するコア数を減ら

し、より多くのコアを持つ他のインスタンスを実行するのと同様またはそれ以上のパフォーマンスを得ることができます。

例えば、r4.4xlarge で実行されている SQL Server Enterprise ワークロードを z1d.3xlarge に移行すると、ライセンスのコア数が減るため、最大 24% の節約を実現できます。ワークロードを r4.16xlarge から z1d.12xlarge に移行する場合も、同じ比率で 4:3 であることから、同じ節約が実現されます。

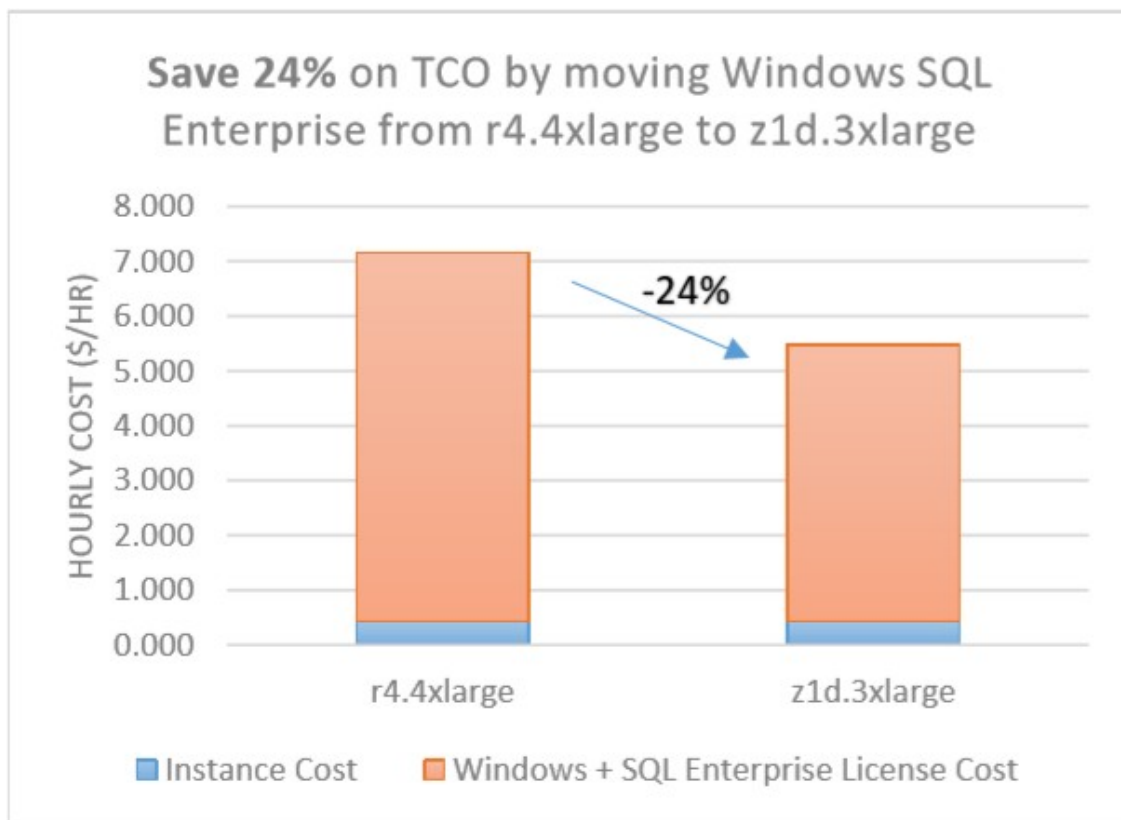


図 4: r4.4xlarge と z1d.3xlarge の SQL Server 間の TCO の比較

## アクティブなレプリカライセンスの排除

クラウドでのコスト最適化のもう 1 つの機会、BYOL モデルと LI モデルの組み合わせを適用することです。一般的なユースケースは、アクティブなレプリカを持つ Microsoft SQL Server Always On 可用性グループです。アクティブなレプリカは、主に次の目的で使用されます。

- レポーティング
- バックアップ
- OLAP バッチジョブ
- HA

上記の 4 つのオペレーションのうち、最初の 3 つは断続的に実行されることがよくあります。つまり、インスタンスを継続的に起動し、それらのオペレーションの実行専用にする必要はありません。従来のオンプレミス環境では、プライマリインスタンスと継続的に同期するアクティブなレプリカを作成する必要があります。つまり、アクティブなレプリカの追加ライセンスを取得する必要があります。

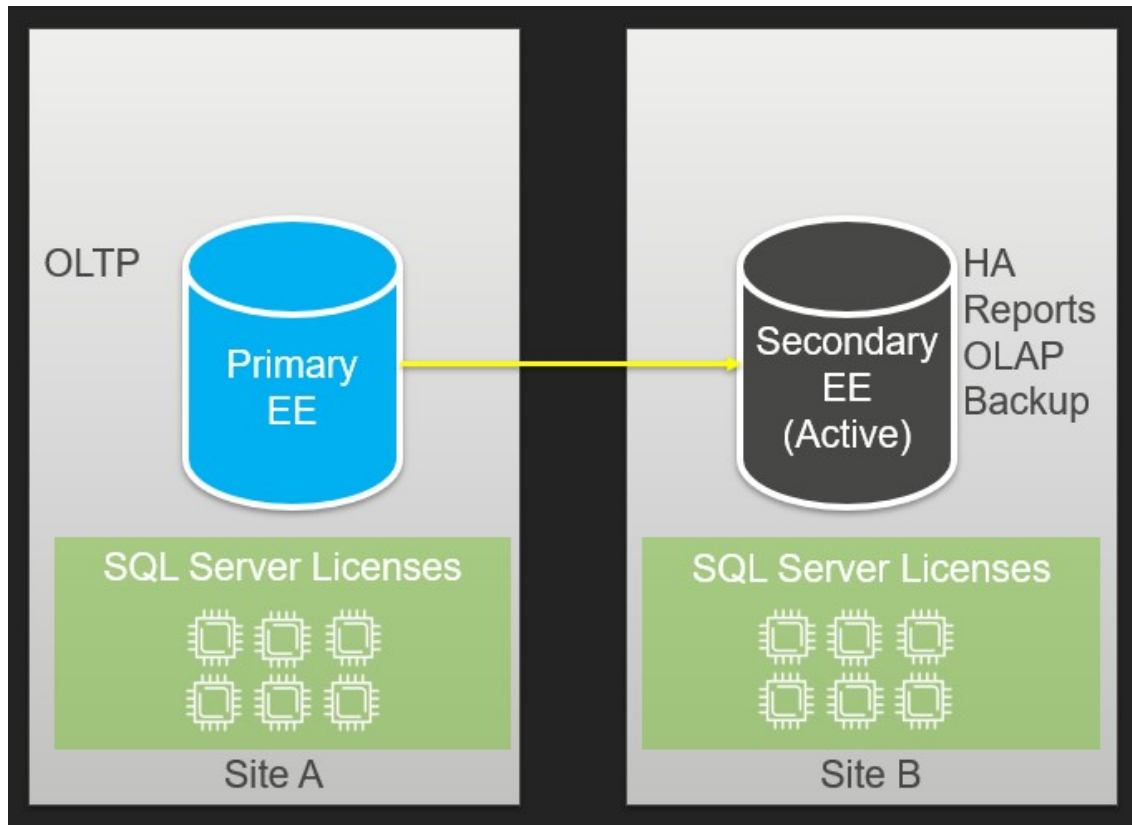


図 5: オンプレミスでの SQL Server アクティブレプリケーション

AWS では、アクティブレプリカをパッシブレプリカに置き換えることで、このアーキテクチャを最適化する機会があり、その役割は HA の目的でのみ引き継がれます。その他のオペレーションは、ライセンス込みを使用して別のインスタンスで実行できます。このオペレーションは、数時間実行してからシャットダウンまたは終了できます。データは、プライマリインスタンスの EBS スナップショットを使用して復元できます。このスナップショットは VSS 対応 EBS スナップショットを使用して作成できるため、プライマリでのパフォーマンスへの影響やダウンタイムは発生しません。

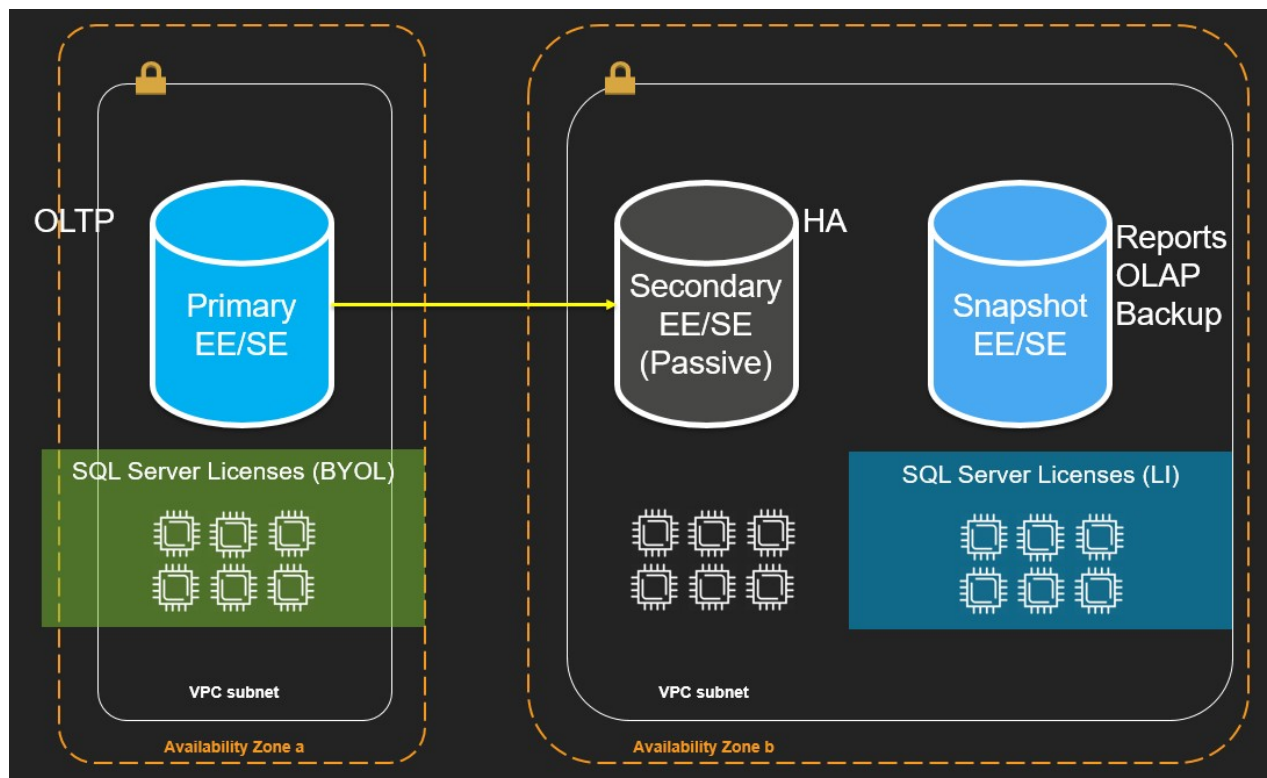


図 6: AWS でのアクティブなレプリカライセンスの排除

このソリューションは、アクティブなレプリカのジョブが低頻度で実行される場合に適用されます。ただし、継続的または高頻度で実行されるジョブのレプリカが必要な場合は、[AWS Database Migration Service](#) (AWS DMS) を使用して、プライマリインスタンスからセカンダリにデータを継続的にレプリケートすることを検討してください。この方法の主な利点は、SQL Server Standard Edition を使用して実行でき、SQL Server Enterprise Edition ライセンスの高コストを回避できることです。

AWS でのライセンスコストを最適化する方法の詳細については、[AWS での Microsoft ライセンシング](#) ページを参照してください。

## 運用上の優秀性

このホワイトペーパーでこれまでに検討した議論のほとんどは、AWS での Microsoft SQL Server のデプロイ時に利用できるベストプラクティスに関連しています。しかしながら、もう 1 つの重要な面は、デプロイ後にこれらのワークロードを運用および維持することです。

一般的な原則として、ベストプラクティスは、障害やインシデントが常に発生すると想定することです。これらのインシデントに対応するために準備し、備えることが重要です。この目標は、次の 3 つの目的で構成されます。

- 異常の観察と検出
- 根本原因の検出
- 問題を解決する行為

AWS では、これらの各目的に合わせてツールとサービスを提供しています。

## 可観測性と根本原因の分析

[Amazon CloudWatch](#) は、AWS リソースやその他のアプリケーションのリアルタイムモニタリングを可能にするサービスです。CloudWatch を使用してメトリクスを収集し、追跡できます。メトリクスとは、リソースやアプリケーションに関して測定できる変数です。

[Amazon CloudWatch Application Insights for .NET and SQL Server](#) は Amazon CloudWatch の機能で、Microsoft SQL Server および .NET アプリケーションの運用性を向上するように設計されています。有効にすると、アプリケーションリソースとテクノロジースタック全体で主要なメトリクスとログを識別して設定します。メトリクスとログを継続的にモニタリングして異常とエラーを検出し、人工知能と機械学習 (AI/ML) を使用して検出されたエラーと異常を関連付けます。工

ラーや異常が検出されると、Application Insights は CloudWatch Events を作成します。トラブルシューティングの円滑化を図るため、検出された問題と関連するメトリクスの異常やログのエラーを示す自動化されたダッシュボードとともに、潜在的な根本原因を示唆する追加的な詳細情報を作成します。

## 平均修復時間の短縮 (MTTR)

Amazon CloudWatch Application Insights によって生成された自動ダッシュボードは、アプリケーションの正常な状態を維持し、アプリケーションのエンドユーザーへの影響を防ぐための迅速な修復アクションを実行するのに役立ちます。OpsItems も作成されるため、[AWS Systems Manager OpsCenter](#) を使用して問題を解決できます。

[AWS Systems Manager](#) は、AWS、オンプレミス、その他のクラウドのインフラストラクチャを表示および管理できるサービスです。OpsCenter は AWS Systems Manager の機能で、解決までの平均時間を短縮するように設計されています。OpsCenter には [Systems Manager Automation](#) ドキュメント (ランブック) も用意されています。これを使用して、問題の解決を完全にまたは部分的に自動化できます。

## パッチ管理

[AWS Systems Manager Patch Manager](#) は包括的なパッチ管理ソリューションで、ネイティブ Windows API と完全に統合されており、Windows Server および Linux オペレーティングシステムと、Microsoft SQL Server を含む Microsoft アプリケーションをサポートします。

AWS Systems Manager Patch Manager は [AWS Systems Manager メンテナンスウィンドウ](#) と統合されているため、予測可能なスケジュールを定義して、ビジネスオペレーションの中断の可能性を回避できます。

また、[AWS Systems Manager 設定コンプライアンス](#)ダッシュボードを使用して、フリート全体のパッチコンプライアンスの状態やその他の設定の不整合をすばやく確認することもできます。

## まとめ

このホワイトペーパーでは、Microsoft SQL Server ワークロードを AWS にデプロイするためのいくつかのベストプラクティスについて説明しました。AWS のサービスを使用して Microsoft SQL Server の機能を補完し、さまざまな要件に対応する方法を確認しました。

AWS では、クラウドで最も幅広いサービスを提供しており、Amazon EC2 は Microsoft SQL Server ワークロードをデプロイするための最も柔軟なオプションです。各ソリューションおよび関連するトレードオフは、特定のビジネス要件に応じて受け入れることができます。

AWS Well-Architected Framework の 5 つの柱（信頼性、セキュリティ、パフォーマンス、コストの最適化、運用上の優位性）が、Microsoft SQL Server ワークロードと、各要件をサポートする AWS サービスに適用可能である場合にご検討ください。

## 寄稿者

本書の執筆に当たり、次の人物および組織が寄稿しました。

- Sepehr Samiei 氏、ソリューションアーキテクト、アマゾンウェブサービス

## ドキュメントの改訂

日付	説明
2020 年 5 月	ドキュメント全体を再改訂し、Microsoft SQL Server ワークロードをサポートする新しい AWS サービスと機能をカバーしました。
2019 年 3 月	z1d インスタンスタイプと EC2 CPU 最適化を使用した総所有コスト (TCO) に関する情報を追加しました。
2018 年 9 月	初版発行