ホワイトペーパー

RED HAT OPENSHIFTで

耐障害性の高いデータベースを実行する

導入のメリット、トレードオフ、およびソリューション

yugabyteDB

Red Hat

目次

はじめに

コンテナ化ワークロードとサービスの管理用プラットフォーム「Kubernetes」は、人気のオープンソースとして、Fortune 500企業で幅広く導入されています。Walmart、Target、eBayなど、多くの企業が現在、このプラットフォームを使用して、ステートレスおよびステートフルなアプリケーションをオンプレミスで、または本番環境に導入されたハイブリッドクラウドとして実行しています。もちろん、新しいテクノロジー全般に言えることですが、Kubernetesでワークロードを実行するときには「産みの苦しみ」があります。しかし、ほとんどのエグゼクティブと開発者は課題よりメリットのほうがはるかに大きいと考えています。

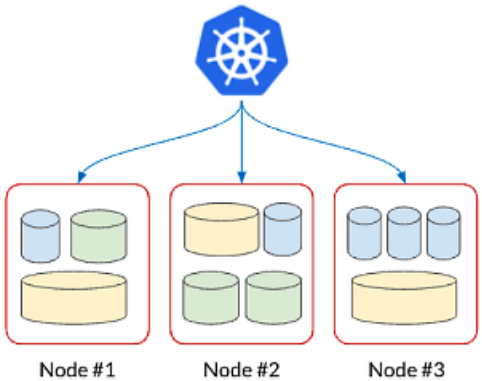
その一方で、Kubernetesエコシステムのデータはステートフルアプリケーションの台頭と共に急速に進化しています。ただし、ステートフルアプリケーションには、アプリケーションの規模、レイテンシー、可用性、およびセキュリティニーズを考慮に入れた新しいデータベースアーキテクチャが必要です。では、どのデータベースアーキテクチャがこれらの課題に最も適切に対処できるかを知るにはどうすればよいのでしょうか。

このホワイトペーパーでは、Kubernetesでデータベースを実行することのメリットと潜在的なトレードオフについて検討します。次に、分散型SQLによってトレードオフを軽減する方法について説明します。分散型SQLは、従来のRDBMSと非SQLデータベースの長所を兼ね備えている新しいクラスのデータベースであり、トランザクションアプリケーションの実行に適しています。最後に、耐障害性に優れ、継続的に利用可能なKubernetes環境を確保するための実際のソリューションをご紹介します。

Kubernetesでデータベースを実行する5つのメリット

Kubernetesでデータベースを実行すべきでしょうか。決定的な答えはありません。しかし、考慮すべき具体的なメリットがあります。Kubernetesでデータベースを実行することが貴社にとって価値がある理由を見てみましょう。

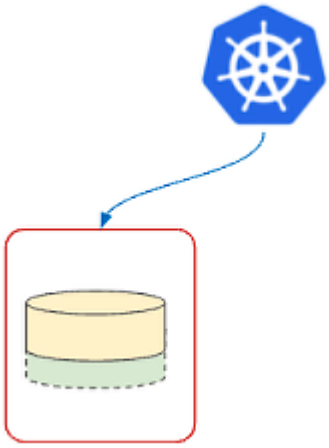
1. **リソース利用率の向上**
   最新アプリケーションでは、多くの企業がマイクロサービスアーキテクチャの採用に移行しています。その結果、このような移行に伴って小規模なデータベースが数多く生み出されています。多くの企業では、これらのデータベースを配置するノードの数が限られているため、このようなデータベースを管理するときに、ノードに最適でない形で割り当てられたデータベースが残ることになります。しかし、Kubernetesを実行すると、基盤となるシステムがそれらのノードへのリソース割り当てを最適化しながら、データベースの最適な配置場所を決定できるようになります。



Kubernetesの最適な活用方法は、多数のデータベースをマルチテナント環境で実行することです。この導入シナリオでは、企業はコストを削減できるだけでなく、同じ種類のデータベースを実行するために必要なノードの数を削減することもできます。これらのデータベースはまた、フットプリント、CPUリソース、メモリー、およびディスク要件がそれぞれ異なります。

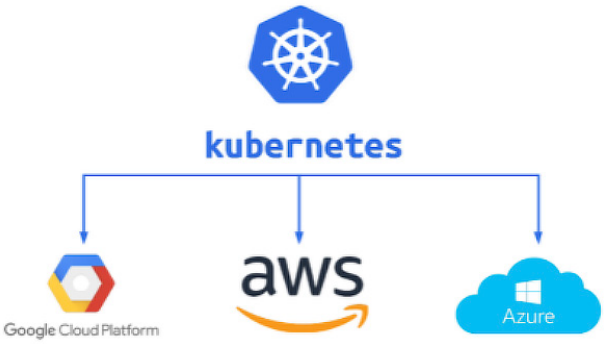
2. **ポッドリソースの動的かつ柔軟なスケーリング**

Kubernetesオーケストレーションプラットフォームでは、ポッドリソースのサイズを動的に変更できます。具体的には、要求の厳しいワークロードに対応できるように、メモリー、CPU、ディスクを変更してデータベースを拡張できます。Kubernetesでは、水平ポッドオートスケーラー（HPA）と垂直ポッドオートスケーラー（VPA）のオペレーターにより、ダウンタイムを発生させずに自動的にスケールアップすることが簡単にできます。ただし、VPAの場合は、ダウンタイムを回避するためにデータベースに複数のインスタンスが必要になることに注意してください。



3. **クラウド、オンプレミス、エッジ間の一貫性と移植性**

企業は、さまざまな場所で一貫した方法でワークロードを構築、導入、管理することを望んでいます。また、必要に応じてワークロードをあるクラウドから別のクラウドに移動する能力を必要としています。しかし、ほとんどの組織は今でも大量のレガシーコードをオンプレミスで実行しており、それらのインストールをクラウドに移行することを検討しています。



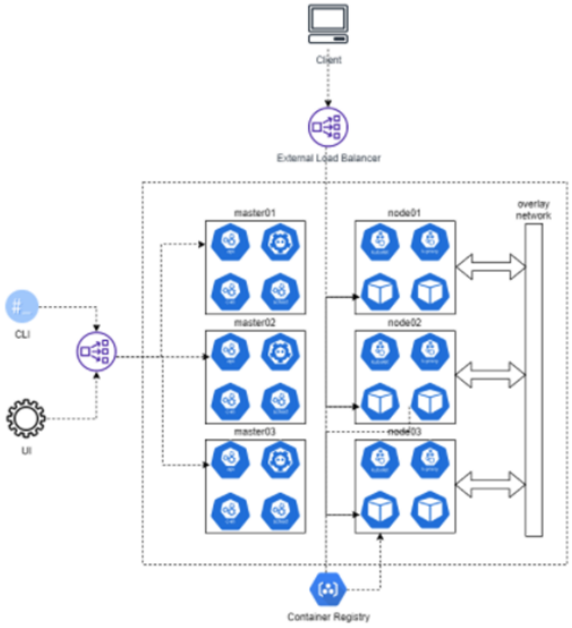
Kubernetesを利用すれば、どこでも一貫した方法でコードとしてのインフラストラクチャを導入できます。これにより、Kubernetesエンジンに割り当てられるリソースの要件を説明する少量のコードを記述するだけで、プラットフォームにリソースの割り当てを任せることができます。データセンターのベアメタルサーバーまたはエッジと同様のコントロール能力がクラウドで手に入ります。

4. **すぐに利用可能なインフラストラクチャオーケストレーション**

Kubernetesでは、ポッドはクラッシュすると自動的に再開します。通常、ポッドはどこでも開始できます。このプラットフォームは、あるポッドのワークロードを別のノードのポッドに移動するよう指示できるからです。これにより、リソース割り当てとリソース利用率を最適化でき特にステートレスワークロードでは非常に効果的です。

Kubernetesポッドにマイクロサービスを展開し、そのポッドの10種類のインスタンスでトラフィックに対応できます。プラットフォームは、ポッドがダウンしようと別のノードに移動しようと意に介しません。そのポッドにはステートがないからです。ただし、データベースでステートフルワークロードを扱うときには、これがより大きな課題になります。つまり、この課題に対処できるようにするための特定のポリシーをKubernetesに設定する必要があります。

たとえば、Kubernetesが従うべきルールをコードで指定できるようにするアンチアフィニティを設定し、「同じデータベースの2つのインスタンスを同じノードに置かない」などのルールを指定できます。これにより、システムにハードウェア障害が発生しても、データベースインスタンスのコピーが複数削除されることはなくなり、同じデータのコピーを複数失う事態を回避できます。



5. **Day 2の運用の自動化**
   Kubernetesでは、バックアップとデータベースソフトウェアのアップグレードを定期的に実行できます。データとデータベースが常に最新の状態に保たれるように、これらの操作を自動化するとよいでしょう。しかもこのアップデートはデータクラスター全体にわたって簡単に実行できます。そのため、セキュリティの脆弱性が見つかった場合、Kubernetesではクラスター全体にわたってその脆弱性にパッチを簡単に適用できます。

   しかし、Kubernetesで従来のRDBMSを併用している場合は、データのコピーを2つ作成することになります。そのため、ポッドを失った場合でも、どこかに別のコピーがあります。それでもなお、これら2つのポッド間でデータを移行し、障害が発生したインスタンスがオンラインに復帰したときにそのインスタンスのデータを再同期する必要があります。Kubernetesでは、これを非同期に行います。従来のRDBMSを使用している場合に自動化された2日目の操作が複雑になることがあるのはそのためです。

   たとえば、データを手動で移行する場合は、クラスターに重い負荷がかかっていないかどうかチェックします。データを別のノードに移動する前に、負荷が軽くなるまで待つ必要があります。一方、データを自動的に移行する場合は、そのようなチェックを慎重に組み込む必要があります。そうしないと、重い負荷によってデータのプライマリコピーが失われた場合、レプリカは実際にないデータがあると認識する可能性があります。これは、2つの異なるデータコピーが存在する可能性があることを意味します。また、データの損失と不一致の可能性も生まれます。

Kubernetesでデータベースを実行することの5つのトレードオフ

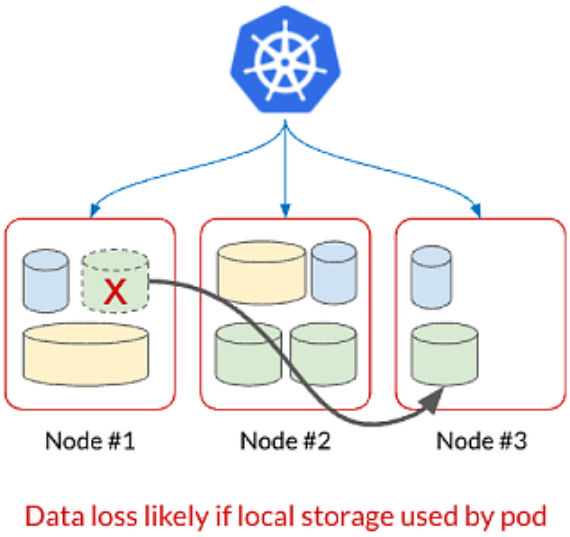
Kubernetesでデータベースを実行すべき理由を説明しました。しかし、新しいテクノロジーに関する意思決定を下すときには、留意すべき潜在的なトレードオフがいくつかあります。

1. **ポッドクラッシュの可能性**
Kubernetesは優れたオーケストレーションツールです。ただし、Kubernetesポッドはプロセスアフィニティがあるため、クラッシュする可能性があります。そのため、ポッドを開始するプロセスがダウンすると、ポッドクラウド全体が消失してしまうことがあります。特定のクエリを実行し、そのクエリによってメモリーが過負荷になるか、設定が不適切になる場合、Kubernetesはポッドをダウンさせることがあります。これはポッド全体のクラッシュにつながる可能性があります。

   ポッドがクラッシュする一般的な理由は、メモリー負荷とOOMキラーです。もう1つの要因は、ポッドの透過的な再スケジューリングです。ノードを追加することはできますが、最適なリソース競合を確保するためにそれらのポッドを移動する必要があります。しかし、異なる種類のストレージを使用する際には別の問題が発生する可能性があります。データベースにストレージをローカル接続すると、パフォーマンスが高速になります。ただし、唯一の問題は、ローカル接続ストレージが新しいノードに存在しないことです。

   例として、以下の図に示すように、ポッドをノード1からノード3に移動するとどうなるか見てみましょう。

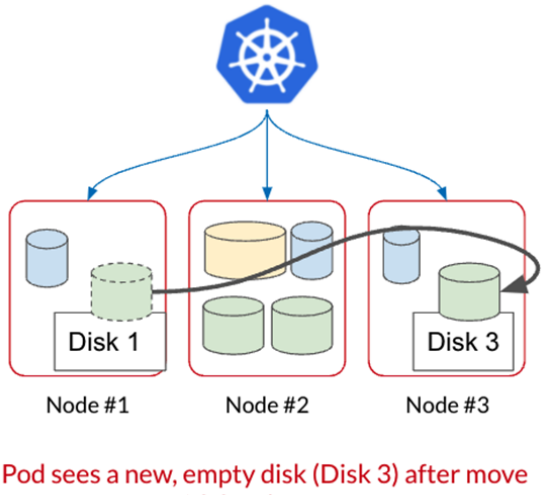
   ノード1にローカル接続ストレージがある場合、そのポッドがノード3に移動すると、ローカル接続ストレージは使用できなくなります。ストレージはそこにあるように見えますが、通常は空です。つまり、データが失われた可能性があります。ネットワーク接続ストレージまたは外部永続ストレージを使用し、新しいポッドを接続すると、この問題を回避できます。



2. **ローカルストレージと外部永続ストレージ**
ローカル接続ストレージは素晴らしいもので、最速のパフォーマンスを提供する単一のドライブまたは一連のドライブです（NVMEドライブなど）。たとえば、レイテンシーが重要な場合は、ローカル接続ストレージが大いに役立ちます。しかし、前のセクションで説明したように、ポッドを移動する場合、ストレージはポッドと一緒に移動しません。つまり、空のドライブから始めることになります。

   外部永続ストレージは、ある種の形態のネットワーク接続ストレージを提供し、Container Storage Interface（CSI）プラグインで簡単に利用できます。クラウドプロバイダーなら簡単に設定できますが、ベアメタルサーバーで設定するのは容易ではありません。外部永続ストレージのメリットは、それがドライブの論理ビューであることです。そのため、固定ドライブを用意する代わりに、ドライブのストレージ容量を指定できる任意の大型ビューが得られます。もう1つのメリットは、ポッドをあるノードから別のノードに移動する際に、ポッドを同じストレージ（仮想ドライブ）とそのインスタンスに簡単に再接続できることです。

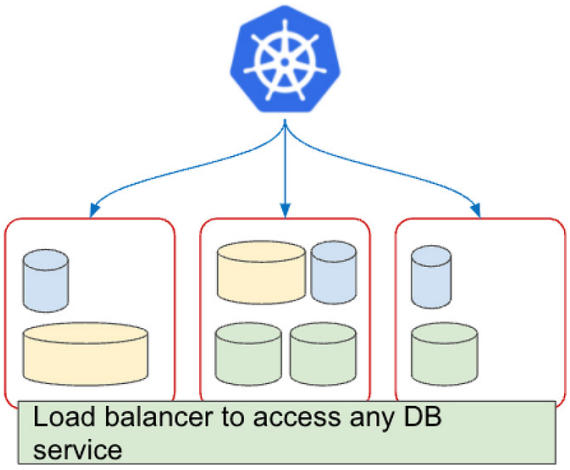
オンプレミス展開はさまざまなソフトウェアソリューションで簡単に管理できます。重要なのは速度と新しいノードの再開のしやすさです。



3. **ロードバランサーに対する潜在的なニーズ**
Kubernetesクラスターでは、ネットワーキングが制限される可能性があります。これにより、アプリケーションを実際のデータベースと同じクラスターに配置しなければならない場合があります。そうでない場合は、ロードバランサーが必要になります。

ロードバランサーを使用すると、クラウド内またはベアメタルサーバー上にあるサービスを外部の顧客に公開できます。これは、パブリッククラウドや一般的なコンテナプラットフォームでは大きな問題ではありません。それらには組み込みのネットワークロードバランサーがあり、取得するパブリックIPアドレスの数以外に制限がほとんどないからです。

考えられる解決策の1つは、すべてのデータベースインスタンスを同じクラスターでホストすることです。ただし、これはアンチパターンである可能性があります。より小規模なクラスターをより多く使用し、局所的な障害のリスクを最小限に抑えることを推奨します。または、別の言い方をすれば、1台のノードまたは一連のノードがクラッシュした場合、構築している分散エコシステム全体でアプリケーションにかかるコストが最小になるようにします。



4. **ネットワーキングが複雑になる可能性**
地理的レプリケーション用にクラスターがセットアップされており、1つは米国東部に、もう1つは米国西部に配置されているとしましょう。自然災害が発生し、クラスター全体がクラッシュした場合に地理的冗長性が確保されるように、それら2つのクラスター間でデータをレプリケートします。その後、自動的に接続して稼働を継続可能なデータの完全コピーをもう1つ用意します。

通常、これらの地域クラスターは別のデータセンターに置かれます。データベースはTCPを介して非常に低いレベルでレプリケートし、効率的なレプリケーションを提供します。ただし、Google Cloudなどの一部のクラウドプロバイダーは、東部と西部間にVPCペアリングを設定してこのような懸念に対処できます。しかし、必ずしもすべてのクラウドプロバイダーがこのサービスを提供しているわけではありません。また、ベアメタルサーバー上で実行している場合は、別の解決策を考え出す必要があります。

そのような解決策の1つがDNSチェイニングです。環境によっては実行が複雑になる場合がありますが、単一のテクノロジーではなく一連のテクノロジーが絡みます。もう1つの解決策は、Istioなどのサービスメッシュを使用することです。サービスメッシュは効果的に機能しますが、TCPではなくHTTP上で動作するため、パフォーマンスの低下を引き起こす可能性があります。オープンソースツールのSubmarinerは、暗号化されたVPNトンネルを使用してさまざまなKubernetesクラスターのオーバーレイネットワークを結び付けるように設計されており、この課題を解決できます。

5. **運用の「落とし穴」**
本番環境のKubernetesでデータベースを実行する場合は、注意すべき「落とし穴」がいくつかあります。

   - アンチアフィニティとポッド停止と見なされる条件を定義する
   - サイドカーのコンセプトを理解する
   - Prometheusなどのツールを使って可観測性を組み込む
   - トラブルシューティングの手順書（メモリー負荷によるオーバーサブスクリプションに起因するクラッシュに備えてバックアップを行う場合に何をすればよいかなど）
   - プライベートイメージレジストリとプールシークレット（アップロードできるユーザー、ダウンロードできるユーザー、適用する権限など）を定義する

多くの企業は、本番環境でKubernetesを実行することに成功しています。しかし、ほとんどの組織は、最初に開発、テスト、ステージングなどの下位本番環境でプラットフォームを実行します。また、本番環境のKubernetesで分散型SQLデータベースを実行している企業が増えています。次のセクションでは、このタイプのデータベースについて詳しく説明します。

分散型SQLデータベースでトレードオフを軽減する

これまで、Kubernetesでデータベースを実行することの多くのメリットと潜在的なトレードオフについて説明してきました。必要なのは、データの継続的な可用性を確保しながらトレードオフを軽減できる分散型SQLデータベースです。このセクションでは、分散型SQLデータベースとは何かを定義し、それをKubernetesで実行したときにトレードオフがどのように軽減されるかを示します。

分散型SQLデータベースとは

分散型SQLデータベースは、サーバーのクラスターに展開される単一の論理リレーショナルデータベースです。このデータベースは、データを複数のサーバーに自動的にレプリケートおよび分散します。これらのデータベースは、強力な一貫性を備えており、クラウドのアベイラビリティゾーンと地理的ゾーンにわたって一貫性をサポートします。

分散型SQLデータベースは少なくとも以下の特性を備えています。

- データとオブジェクトにアクセスして操作するためのSQL API
- クラスターの複数のノードへのデータの自動分散
- 強く一貫した方法でのデータの自動レプリケーション
- クライアントが基盤となるデータ分散について知らなくても済むようにするための分散クエリ実行のサポート
- 分散ACIDトランザクションのサポート

分散型SQLデータベースを選ぶ理由

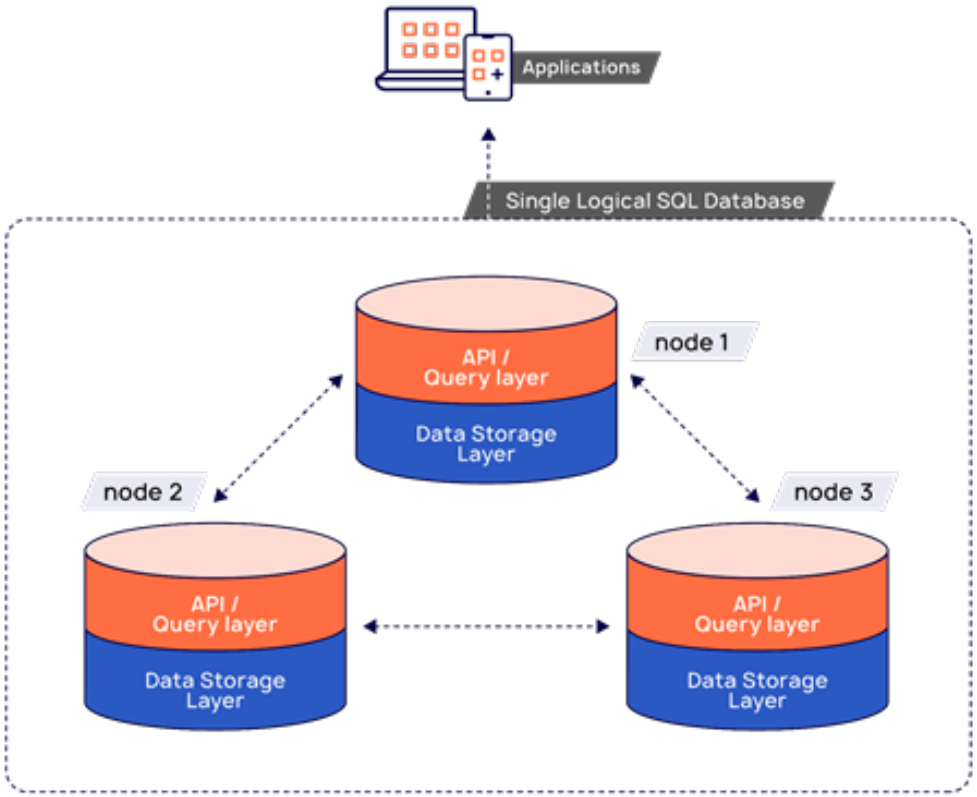
従来の記録システムはビジネスイノベーションの圧力を受けています。これにより、企業はITコストを削減し、コンプライアンスによってリスクを軽減しながら、高価値のアプリケーションとサービスを提供する必要に迫られています。

ただし、こうしたアプリケーション（例えば、マイクロサービス、クラウドネイティブアプリケーション、エッジ、IoTワークロード）には以下のような特性を備えた新しいクラスのデータベースが必要です。

- **耐障害性に優れ、継続的に利用可能：**高速フェールオーバーにより、ノード、ゾーン、リージョン、データセンターで障害が発生したときやシステムメンテナンスを実行しているときでも重要なサービスを継続的に利用できる
- **水平方向に拡張可能：**運用チームが、重い負荷がかかっているときでもクラスターにノードを追加するだけで、ダウンタイムなしで簡単にスケールアウトでき、負荷が軽くなったときにスケールバックできる
- **地理的に分散：**オペレーターが同期および非同期のデータレプリケーションと地理的分割を活用し、地理的に分散された構成でデータベースを展開できる
- **SQLとRDBMS機能の互換性：**開発者がクラウドネイティブシステムの水平方向の拡張性と従来のRDBMSのACID保証および強力な一貫性のいずれか一方のみを選ぶ必要がなくなる
- **ハイブリッドおよびマルチクラウド対応：**組織が場所を選ばずデータインフラストラクチャを展開して実行し、特定のクラウドプロバイダーにロックインされるのを回避できる

分散型SQLデータベースのアーキテクチャ

分散型SQLデータベースは、従来のRDBMSの長所とクラウドネイティブなデータベース機能を兼ね備えています。単一の論理SQLデータベースの一部として2レイヤーアーキテクチャを備えています。



SQLクエリレイヤー

分散型SQLデータベースは、アプリケーションがリレーショナルデータをモデリングし、それらのリレーションに関連するクエリを実行するためのSQL APIを備えています。クエリは、データベースクラスターの複数のノードに自動的に分散されます。

分散型データストレージレイヤー

分散型SQLデータベースのデータ（インデックスを含む）は、クラスターの複数のノードに自動的に分散（またはシャーディング）されます。そのため、1つのノードがハイパフォーマンスと高可用性のボトルネックになることはありません。

強力なSQL APIレイヤーをサポートするには、基盤となるストレージレイヤーが、クラスターのすべてのノードにわたる強く一貫したレプリケーションに基づいて構築されている必要があります。これは、障害発生時の可用性を保証するためにデータベースへの書き込みが複数のノードで同期的にコミットされることを意味します。

そして最後に、データベースストレージレイヤーは、複数のノードにある複数の行にわたってトランザクションの調整が必要になる分散ACIDトランザクションをサポートします。

分散型SQLデータベースがKubernetesのトレーオフを軽減する方法

分散型SQLデータベースは、ノードのクラスターとして展開された単一の論理データベースとして機能します。つまり、データベースクラスターはシャーディング、レプリケーション、ロードバランシング、データ分散を実行します。したがって、分散型SQLデータベースは、ポッド、ノード、または基盤となるインフラストラクチャに障害が発生した場合でも、データベースを稼働させ続けます。データベースクラスターは、障害を検出してそれに対処し、データまたはアプリケーションアクセスを一切失うことなく復旧できます。

分散型SQLデータベースは、アプリケーションを結び付けるための、拡張性と耐障害性に優れたデータストアでもあります。分散型SQLデータベースは、ポッドが新しいノードに移動した後、ポッド間でデータを移行します。これは、いかなる形のオペレーターの介入もなしにバックグラウンドで行われます。

YugabyteDB：耐障害性の高いKubernetesワークロード向けのクラス最高の分散型SQL

YugabyteDBは、トランザクションアプリケーション向けのクラウドネイティブな分散型SQLデータベースです。このデータベースは100%オープンソースで、Kubernetesアプリケーションワークロードを実行するときの可用性と耐障害性の課題を解決できるように設計されています。

このデータベースは、高可用性を確保するためにレプリカを使用し、Raftプロトコルを使用することで同期をサポートします。その他の機能には、拡張性を確保するための分割（タブレットと呼ばれます）があり、タブレット間トランザクションの場合は、2段階コミットプロトコルも実装されます。

YugabyteDBは、ユーザーの介入なしにSQLテーブルをタブレットに分割します。また、データ損失ができる限り発生しないように、タブレットレプリカを構成済みの障害ドメインに自動的に分散します。この動作は、ユーザーの障害ドメイン、レプリケーション係数、障害ドメインに対するデータベースアフィニティの設定によって変化する場合があります。

導入の柔軟性

YugabyteDBは、パブリック、プライベート、およびハイブリッドクラウド環境、VM、コンテナ、またはベアメタルで動作します。組織は任意のKubernetes環境にこのデータベースを導入できます。このデータベースは、スムーズなエクスペリエンスを実現するマルチクラウドフルマネージド型のサービスとしてのデータベース（DBaaS）として利用することもできます。YugabyteDBは、分散型SQLデータベースの中で最も広範なレプリケーションおよび地理的分散オプションを提供します。

ハイパフォーマンス

YugabyteDBは、Kubernetesの高スループット・低レイテンシートランザクションに対処できます。毎秒100万トランザクションと数千の同時接続に拡張できることが本番環境で実証されています。

運用のシンプルさ

組織は、YugabyteDBのセルフマネージド型またはフルマネージド型のDBaaSサービスを使用して、エッジおよびクラウドでの運用をシンプル化できます。このデータベースはまた、その他のデータソースまたはシンクと統合されます。これにより、データエンジニアは機械学習、分析、長期ストレージ、災害復旧のためのパイプラインを構築できるようになります。

PostgreSQLとの互換性

YugabyteDBはPostgreSQLとワイヤーだけでなく、コードでも互換性があります。このデータベースはまた、トリガー、関数、ストアドプロシージャ、強力なセカンダリインデックスなどの包括的で高度なRDBMS機能一式を備えています。このため、開発者は使い慣れたインターフェイスとPostgreSQL互換のフレームワーク、アプリケーション、ドライバー、ツールからなる充実したエコシステムを活用して即座に生産性を高められます。

セキュリティ

YugabyteDBは、セキュリティを念頭に置いてゼロから構築されており、組織がより分散化された環境でも堅牢なセキュリティ体制を維持することを可能にします。YugabyteDBは、保存中および移動中のデータの暗号化、データベースレイヤーでのマルチテナンシーのサポートとテナントごとの暗号化、コンプライアンスを確保するための地域局所性、地理的なアクセス制御の管理などの機能を備えています。

YugabyteDBとRed Hat OpenShift：エンタープライズ規模のクラウドネイティブな耐障害性

YugabyteDBはRed Hat® OpenShift®で使用できます。Red Hat OpenShiftは、クラウドネイティブなアプリケーションを導入・管理するための主要なエンタープライズKubernetesプラットフォームです。Red Hat OpenShiftは、Kubernetesコンテナ管理プロジェクトを基盤に構築されたソフトウェア製品ですが、大企業にとって重要な生産性機能とセキュリティ機能を強化しています。Kubernetesではさまざまなタスクを実行できるものの、ユーザーはなお、ネットワーキング、アクセス許可およびロードバランシング、ストレージ、モニタリング、ロギングなどのその他のコンポーネントを組み込む必要があります。Red Hat OpenShiftは、Kubernetesと共にこれらのコンポーネントをKubernetesの中核として提供します。Kubernetesは単体では不十分だからです。

Red Hat OpenShiftは、任意のクラウドやオンプレミスで実行される既存のモダナイズされたクラウドネイティブなアプリケーションを管理するための、一貫性のあるアプリケーションプラットフォームです。また、任意のインフラストラクチャをカバーする共通の抽象レイヤーを提供し、開発者と運用チームの両方が共通の方法でアプリケーションをパッケージ化、導入、および管理できるようにします。これにより、チームは自信を持ってRed Hat OpenShiftにYugabyteDBを展開できます。YugabyteDBのコンテナイメージとOperatorはRed Hatの認証を受けているからです。そのため、どちらのコンポーネントもプラットフォーム上で動作し、1〜2日目の効率的な運用を可能にするように適切に統合されています。

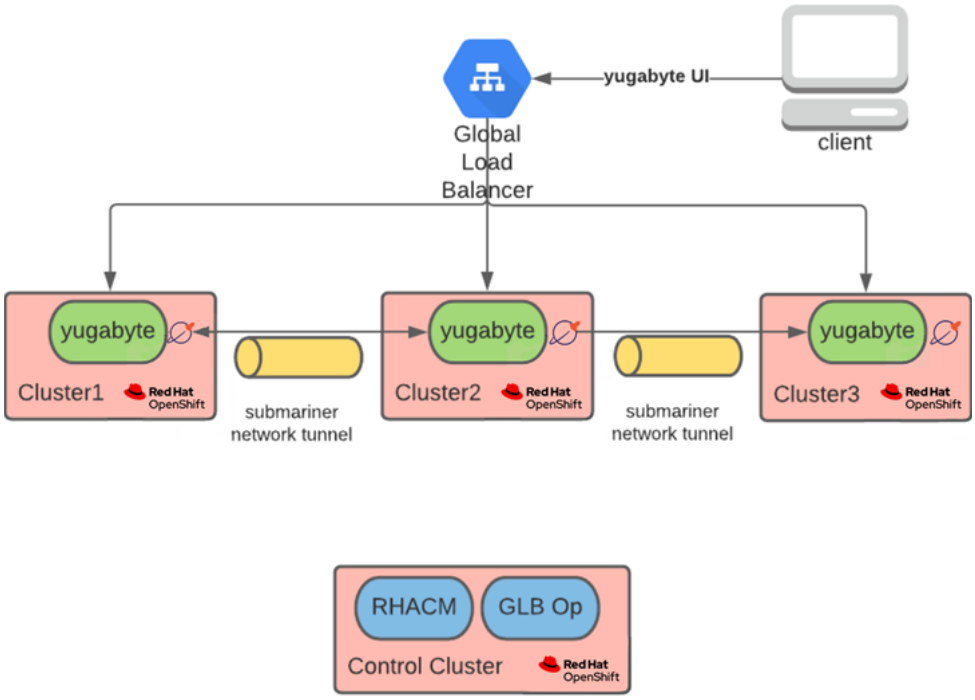
「どこでも実行できる」分散型ステートフルワークロード

Red Hat OpenShiftでYugabyteDBを実行することの大きなメリットの1つは、地理的に分散されたステートフルワークロードが保証されることです。以下の図は、このようなアーキテクチャの実態を示しています。

一番上には、YugabyteDB UIに接続を誘導するグローバルロードバランサーがあります。その下には3つのRed Hat OpenShiftクラスターがあり、それぞれにYugabyteDBインスタンスが展開されています。これらのインスタンスは、Submarinerで実装されたネットワークトンネルを介して相互に通信できます。

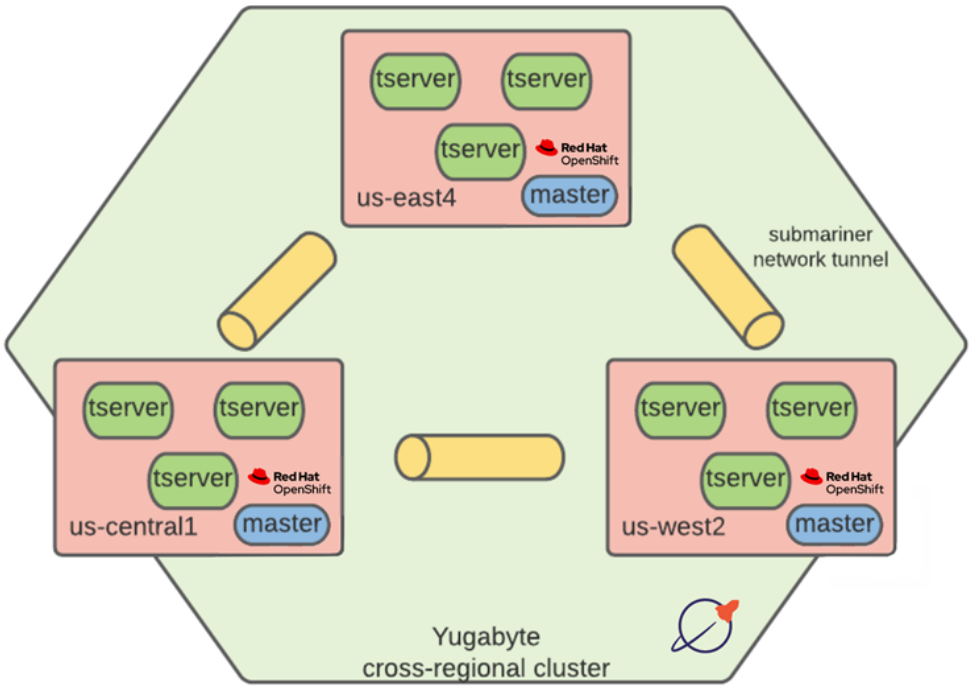
最後に、図の一番下にあるRed Hat Advanced Cluster Manager（RHACM）は、コントロールクラスター内にインストールされており、その他のクラスターとグローバルロードバランサーオペレーターを作成するために使用されます。グローバルロードバランサーオペレーターは、図の一番上にあるグローバルロードバランサーを簡単に構成できるようにします。

各クラスターは、パブリッククラウドプロバイダーの異なるリージョンにあります。



YugabyteDB展開を拡大すると、各クラスターに3台のタブレットサーバーと1台のマスター（メタデータサーバー）があることがわかります。これらは一体となって論理的なYugabyteDBインスタンスを形成します。

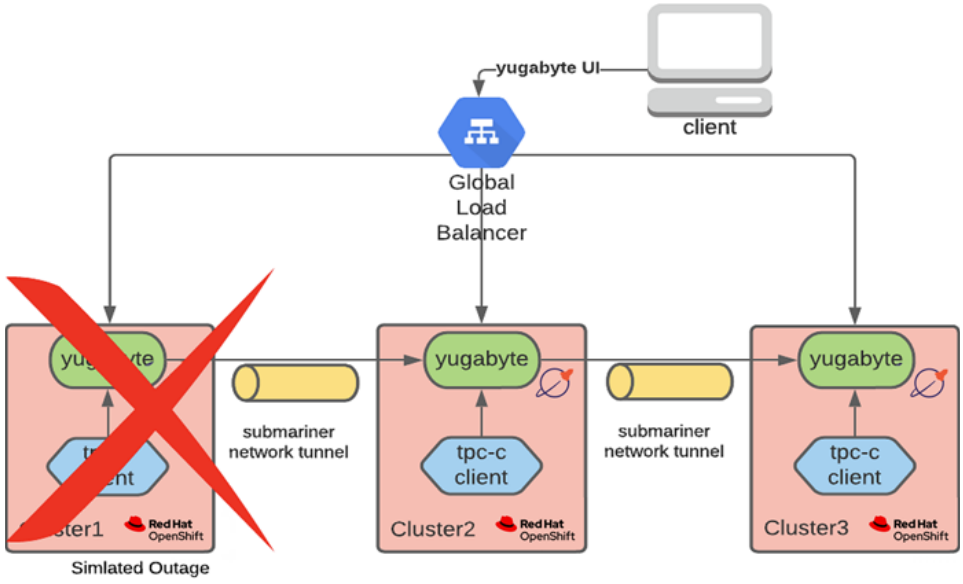
この地理的に分散されたステートフルに関するドキュメントに記載されているYugabyteとRed Hatの間で行われた統合作業で示されているように、この種の展開は本番環境でも有用であることが負荷テストの結果によって実証されています。



データ損失ゼロと継続的な可用性

Red Hat OpenShiftでYugabyteDBを実行することのもう1つのメリットは、大規模なシステム停止や自然災害が発生したときのデータ損失ゼロと継続的な可用性です。

たとえば、以下の図では、TPC-Cテストの実行中にインバウンドまたはアウトバウンドトラフィックを阻止することにより、1つのリージョンのネットワークを分離しています。



この災害をシミュレートしているとき、残存するTPC-Cクライアントにいくつかのエラーがありました。基本的に、いくつかの実行中のトランザクションは拒否されるか、正常に完了しませんでした。しかし、YugabyteDBはすべてのタブレットリーダーを正常なインスタンスに移動しました。

システムは、人間の介入を一切必要とせずに災害に対処しました。

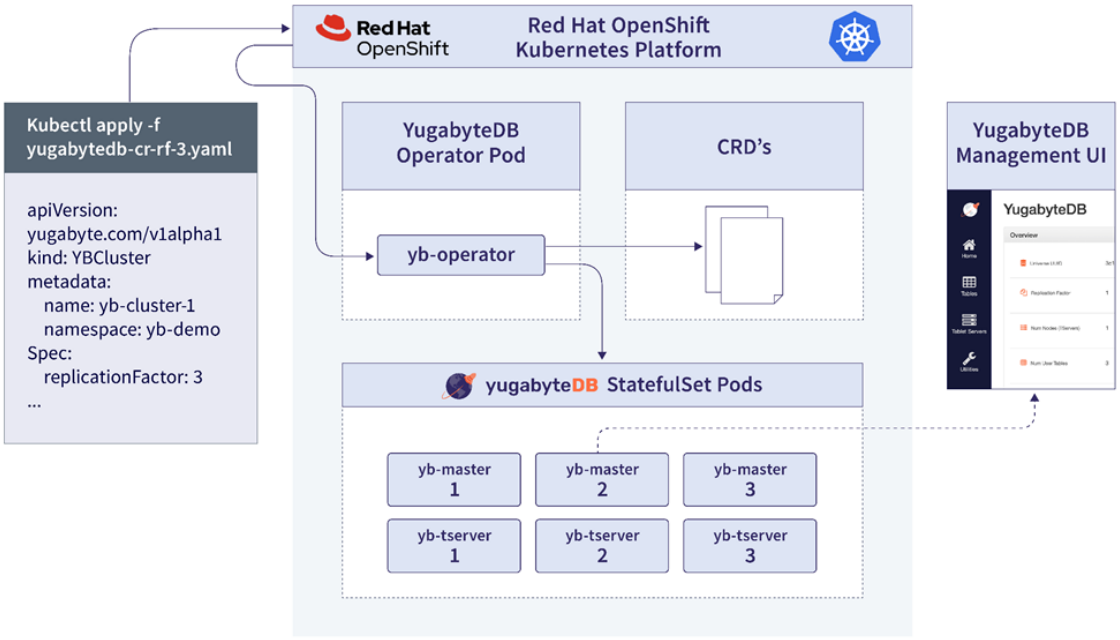
分離されたリージョンへの接続が回復したとき、実行中のTPC-Cクライアント内に問題はありませんでした。YugabyteDBは、新たに利用可能になったタブレットサーバーにタブレットリーダーを戻すことでデータベースを再バランスしました。ここでもまた、人間の介入は必要ありませんでした。

このシミュレーションの実行中、システムでデータの損失は発生せず（RPO 0）、システムを利用できなかった時間はほんのわずかでした（秒単位で測定されたRTO）。

Red Hat OpenShiftでYugabyteDBを始める

YugabyteDBは、Red Hat OpenShift上でのスケールアウトRDBMSとインターネット規模のOLTPワークロードの両方の実行をサポートします。これにより、顧客はこれらのワークロードをエンタープライズグレードのKubernetesに移行できます。YugabyteDB Operatorにより、開発者はステートレスアプリケーションで使用するようになったものと同じクラウドネイティブな手順（CI/CDパイプラインを使用してワークロードを拡張し、そのライフサイクルを管理するなど）を使用し、Red Hat OpenShiftでYugabyteDBクラスターを実行できます。YugabyteDB Kubernetes Operatorは、Red Hat OpenShift環境のOperatorHubにあります。

以下の図は、YugabyteDB Operatorを使用したRed Hat OpenShiftへのYugabyteDBクラスターの展開に関連するコンポーネントの概要を示しています。



YugabyteDBクラスター展開は、yb-tserverとyb-masterの2つの分散型サービスで構成されます。yb-tserverサービスは、アプリケーションデータを保存し、クライアントのリクエストに対応します。yb-masterは、システムメタデータ（table-to-shard-to-nodeマッピングを含む）を管理し、再バランシングなどのバックグランド操作を実行する軽量のサービスです。YugabyteDBクラスターのコンポーネントの詳細については、こちらをご覧ください。

Red Hat OpenShiftは管理されたKubernetesクラスターを提供します。Operator Lifecycle Manager（OLM）は、YugabyteDB OperatorポッドとKube APIに登録されているCRDのライフサイクルを管理します。YugabyteDB Operatorは、カスタムリソースybclusters.yugabyte.comのインスタンスを作成する際に、レプリケーション係数や望ましいポッド数などの指定された属性を使用して必要なステーフルセットポッドを作成します。YugabyteDB Operatorは、YugabyteDB管理コンソールを表示するためにLoadBalancerなどの追加のサービスをブートストラップします。

上の図に示されているYugabyte Platformは、Red Hat OpenShift環境にも導入できます。Yugabyte Platformは、セルフマネージド型のDBaaSを大規模に提供するためのシンプルさとサポートを提供します。Yugabyte Platformは、YugabyteDBクラスターをプロビジョニングし、そのライフサイクルを管理するためのYugabyte推奨のメカニズムです。

以下は、Red Hat OpenShiftでYugabyteDBを始めるための段階的な手順です。

1. YugabyteDB Operatorを使用してYugabyte Platformを導入する
2. Yugabyte Platform内でRed Hat OpenShiftを構成する
3. Yugabyte Platformを使用して導入環境を作成および管理する

まとめ

Kubernetesは、ますますクラウドネイティブ化する世界のニーズに応えるための、企業のアプリケーションの構築方法および導入方法におけるパラダイムシフトです。このような環境のあらゆるアプリケーションに対応できる万能のデータベースリファレンスアーキテクチャは存在しません。アプリケーションの要件と関連するトレードオフに応じて、企業は自社のニーズを満たすさまざまなトポロジーを選択し、ニーズが変化したときにトポロジーを変更します。分散型SQLデータベースは、クラウドとKubernetesの両方の環境でアプリケーションを実行するための強力で汎用的なデータレイヤーを提供します。

Yugabyte, Inc.について

Yugabyteは、グローバルでクラウドネイティブなアプリケーションを構築するための、オープンソースのハイパフォーマンス分散型SQLデータベースであるYugabyteDBを提供している企業です。YugabyteDBは、SQLクエリの柔軟性、ハイパフォーマンス、クラウドネイティブな機敏性によってビジネスクリティカルなアプリケーションをサポートし、企業が複雑なインフラストラクチャ管理ではなくビジネスの成長に専念できるようにします。YugabyteDBは、サイバーセキュリティ、金融市場、IoT、小売、Eコマースなどの業種のグローバル企業から信頼されています。Yugabyteは、以前にFacebookとOracleでエンジニアを務めていた人々によって2016年に設立され、Lightspeed Venture Partners、8VC、Dell Technologies Capital、Sapphire Venturesなどの企業から支援を受けています。

Red Hat, Inc.について

Red Hatは、エンタープライズオープンソースソフトウェアソリューションの世界的な大手プロバイダーで、コミュニティ駆動型のアプローチを活用して信頼性とパフォーマンスに優れたLinux、ハイブリッドクラウド、コンテナ、Kubernetesテクノロジーを提供しています。Red Hatは、顧客が新規および既存のITアプリケーションを統合したり、クラウドネイティブなアプリケーションを開発したり、当社の業界をリードするオペレーティングシステムで標準化したり、複雑な環境を自動化、保護、および管理したりするのを支援しています。受賞歴のあるサポート、トレーニング、およびコンサルティングサービスを提供しているRed Hatは、Fortune 500企業にとって信頼できるアドバイザーです。Red Hatは、クラウドプロバイダー、システムインテグレーター、アプリケーションベンダー、顧客、オープンソースコミュニティの戦略的パートナーとして、組織がデジタルの未来に備えるのを支援できます。

連絡先

www.yugabyte.com | contact@yugabyte.com

github.com
/yugabyte

yugabyte.com
/slack

twitter.com
/yugabyte

youtube.com
/yugabyte

linkedin.com/company
/yugabyte