
記事

[Toshihiko Minamoto](#) · 2020年8月11日 44m read

Google Cloud Platform (GCP)向けのIRISサンプルアーキテクチャ

Google Cloud Platform (GCP) は、IaaS (サービスとしてのインフラストラクチャ) 向けの機能性豊かな環境をクラウドとして提供しています。最新の InterSystems IRIS データプラットフォームなど、InterSystems の全製品に完全に対応していますが、あらゆるプラットフォームやデプロイメントモデルと同様に、パフォーマンス、可用性、運用、管理手順などの環境に関わるすべての側面が正しく機能するように注意を払う必要があります。この記事では、こういった各分野の詳細について説明しています。

以下の概要と詳細は Google が提供しているものであり、[こちら](#)から参照できます。

概要

GCP リソース

GCP は、コンピュータやハードドライブといった一連の物理アセットと、仮想マシン (VM) といった仮想リソースから構成されており、世界中の Google データセンターに保管されています。それぞれのデータセンターの場所は 1 つのグローバルリージョンにあります。そして、それぞれのリージョンはゾーンのコレクションであり、これらのコレクションはリージョン内で互いに分離されています。各ゾーンには、文字識別子とリージョン名を合わせた名前が付けられています。

このようなリソースの分散には、障害発生に備えて冗長性を得たり、リソースをクライアントに近い場所に配置することでレイテンシを軽減できたりなど、さまざまなメリットがあります。また、このような分散によって、リソースをどのように合わせて使用できるかに関するルールも導入することができます。

GCP リソースへのアクセス

クラウドコンピューティングでは、物理的なハードウェアとソフトウェアがサービスとなります。これらは基盤リソースへのアクセスを提供するサービスです。InterSystems IRIS ベースのアプリケーションを GCP で開発する場合、必要としているインフラストラクチャを得られるようにこれらのサービスを組み合わせ、コードを追加することで、構成しようとしているシナリオを実現することができます。利用可能なサービスの詳細は、[こちら](#)を参照してください。

プロジェクト

割り当てて使用する GCP は、プロジェクトに属するものである必要があります。プロジェクトは、設定、権限、およびアプリケーションを説明するその他のメタデータで構成されます。単一のプロジェクト内のリソースは、リージョンとゾーンのルールに従って、内部ネットワークを介した通信などによって簡単に連携することができます。各プロジェクトに含まれるリソースは、プロジェクトの境界で分離されたままであるため、外部ネットワーク接続を介してしか相互接続することはできません。

サービスとの対話

GCP でサービスとリソースを操作するには、基本的に 3 つの方法があります。

コンソール

Google Cloud Platform Console は、Web ベースのグラフィカルユーザーインターフェースを提供しており、ユーザーは、それを使用して GCP のプロジェクトとリソースを管理することができます。GCP Console を使用する場合は、新規プロジェクトを作成するか既存のプロジェクトを選択すると、作成するリソースをプロジェクトの文脈で使用できます。複数のプロジェクトを作成できるため、ユーザーに意味のある方法で、プロジェクトを使用して分類することができます。たとえば、リソースにアクセスできるユーザーを特定のチームメンバーに限定する場合は新しいプロジェクトを開始し、すべてのチームメンバーは引き続き別のプロジェクトのリソースにアクセスすることができます。

コマンドラインインターフェース

ターミナルウィンドウで作業する場合、Google Cloud SDK の `gcloud`

コマンドラインツールを使用して、必要なコマンドにアクセスすることができます。gcloud ツールでは、開発ワークフローと GCP リソースの両方の管理を行えます。gcloud のリファレンスについては、[こちら](#)をご覧ください。

GCP には、Cloud Shell という、ブラウザで使用する GCP 向けのインタラクティブシェル環境も用意されています。Cloud Shell には、GCP Console からアクセスできます。以下に、Cloud Shell の特徴を示します。

- 一時的な Compute Engine 仮想マシンインスタンス
- Web ブラウザからインスタンスへのコマンドラインアクセス
- ビルトインのコードエディタ
- 5 GB の永続ディスクストレージ
- プレインストールされた Google Cloud SDK およびその他のツール
- Java、Go、Python、Node.js、PHP、Ruby、および .NET 言語のサポート
- Web プレビュー機能
- GCP Console プロジェクトとリソースへのアクセスの組み込み認証

クライアントライブラリ

Cloud SDK には、リソースを簡単に作成して管理できるクライアントライブラリが含まれています。GCP クライアントライブラリは、主に 2 つの目的で API を公開しています。

- アプリ API はサービスへのアクセスを提供します。アプリ API は、Node.js や Python といったサポート対象の言語用に最適化されています。ライブラリは、サービスのメタファーを軸に設計されているため、ユーザーはサービスをより自然に操作し、より少ないボイラープレートコードを記述することができます。ライブラリは、認証と承認のヘルパーも提供しています。詳細は、[こちら](#)を参照してください。
- 管理 API は、リソース管理機能を提供します。たとえば、独自の自動化ツールを構築する場合、管理 API を使用できます。

また、Google API クライアントライブラリを使用して、Google マップ、Google ドライブ、および YouTube といった製品用の API にアクセスすることもできます。GCP クライアントライブラリの詳細は、[こちら](#)を参照してください。

InterSystems IRIS サンプルアーキテクチャ

この記事の一部として、アプリケーション固有のデプロイメントの出発点として、GCP 向けの InterSystems IRIS サンプルデプロイメントを提供しています。

デプロイメントの可能性には多数ありますが、これらのサンプルをガイドラインとしてご利用ください。このリファレンスアーキテクチャでは、規模の小さなデプロイメントから非常にスケーラブルなワークロードまで、コンピューティングとデータの両方の要件に対応する非常に堅牢なデプロイメントオプションを紹介しています。

このドキュメントでは、高可用性と災害復旧に関するオプション、そしてその他の推奨システム運用について説明しています。組織の標準的なやり方やセキュリティポリシーに応じて変更してください。

InterSystems では、ユーザー固有のアプリケーションについて、GCP ベースの InterSystems IRIS デプロイメントに関するご相談またはご質問をお受けしています。

サンプルリファレンスアーキテクチャ

以下のサンプルアーキテクチャでは、キャパシティと機能を高めるさまざまな構成を提供します。小規模な開発、本番、大規模な本番、およびシャードクラスタを使用した本番を検討してください。開発作業用の規模の小さな構成から、ゾーン全体に適した高可用性とマルチリージョン災害復旧を備えた非常にスケーラブルなソリューションへと構成が増大していく様子を確認できます。さらに、SQL

クエリの超並列処理によるハイブリッドワークロードに対する InterSystems IRIS データプラットフォームの新しいシャーディング機能を使用するサンプルアーキテクチャも用意されています。

小規模開発の構成

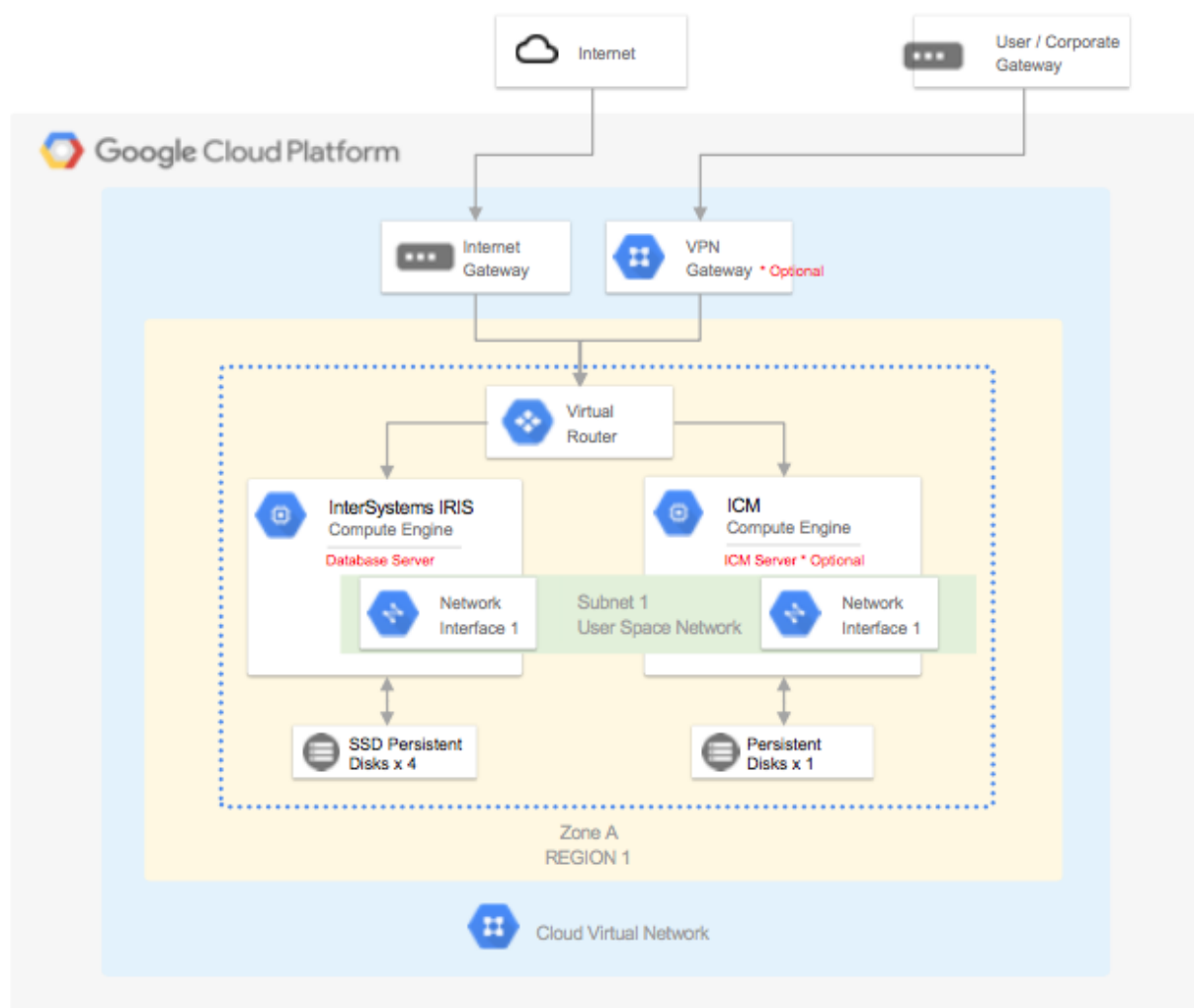
この例では、最小構成を使用して、開発者数最大10人と100GBのデータに対応できる小規模な開発環境を示します。仮想マシンのインスタンスの種類を変更し、永続ディスクのストレージを適切に増加するだけで、より多くの開発者とデータ量を簡単にサポートできるようになります。

これは、開発作業をサポートし、InterSystems IRIS の機能性と、必要であれば Docker コンテナの構築とオーケストレーションに慣れる上で十分な構成です。小規模な構成では通常、データベースミラーリングによる高可用性を使用することはありませんが、高可用性が必要な場合にはいつでも追加することができます。

小規模な構成のサンプル図

以下の図 2.1.1-a のサンプル図は、図 2.1.1-b のリソーステーブルを示します。含まれているゲートウェイは単なる例であり、組織の標準的なネットワークに合わせて調整できます。

Figure-2.1.1-a: Sample Small Development Architecture



以下の GCP VPC プロジェクト内のリソースは、最低限の小規模構成としてプロビジョニングされています。GCP リソースは必要に応じて追加または削除することができます。

小規模構成の GCP リソース

??????????????? GCP ???????????????????

VPC への不要なアクセスを防止するには、適切なネットワークセキュリティとファイアウォールのルールを検討する必要があります。Google は、これに取り掛かるためのネットワークセキュリティに関するベストプラクティスを[こちら](#)で提供しています。

注意: VM インスタンスが GCP サービスにアクセスするには、パブリック IP アドレスが必要です。このやり方では懸念を生じてしまう可能性があります、Google

は、ファイアウォールのルールを使用して、これらの VM インスタンスへの受信トラフィックを制限することを推奨しています。

セキュリティポリシーで VM

インスタンスが完全に内部化されていることが要求されている場合、ネットワークと対応するルートに手動で NAT プロキシを設定し、内部インスタンスがインターネットにアクセスできるようにする必要があります。 SSH を使用して、完全に内部化された VM

インスタンスを直接接続することはできないことに注意しておくことが重要です。

そのような内部マシンに接続するには、外部 IP アドレスを持つ踏み台インスタンスをセットアップしてから、それを通してトンネルをセットアップする必要があります。 外部に面する VPC へのエントリポイントを提供する踏み台ホストをプロビジョニングすることができます。

踏み台ホストの詳細は、[こちら](#)を参照してください。

本番構成

この例では、InterSystems IRIS データベースミラーリング機能を組み込んで高可用性と災害復旧をサポートする本番構成の例として、より大規模な構成を示しています。

この構成には、自動フェイルオーバーを行うための、region-1 内で 2 つのゾーンに分割された InterSystems IRIS データベースサービスの同期ミラーペアと、GCP リージョン全体がオフラインになるという稀なイベントにおいて災害復旧を行うための、region-2 の 3 番目の DR 非同期ミラーメンバーが含まれます。

InterSystems Arbiter と ICM サーバーは、回復力を得るために、別の 3 番目のゾーンにデプロイされています。

サンプルアーキテクチャには、Web

対応アプリケーションをサポートするためのオプションとして、一連の負荷分散された Web

サーバーも含まれています。 InterSystems Gateway を備えたこれらの Web

サーバーは、必要に応じて個別に拡張することができます。

本番構成のサンプル図

以下の図 2.2.1-a のサンプル図は、図 2.2.1-b のリソーステーブルを示します。

含まれているゲートウェイは単なる例であり、組織の標準的なネットワークに合わせて調整できます。

以下の GCP VPC

プロジェクト内のリソースは、シャードクラスタをサポートするための最低推奨事項として推奨されています。

GCP リソースは必要に応じて追加または削除することができます。

本番構成の GCP リソース

???????????? GCP ??????????????

大規模な本番の構成

この例では、InterSystems IRIS の機能を拡張することで、InterSystems Cache プロトコル (ECP) を使用したアプリケーションを導入してユーザーの大規模な水平スケーリングも提供できる、大規模な構成を示しています。この例にはさらに高レベルの高可用性が含まれており、データベースインスタントのフェイルオーバーが発生した場合でも ECP クライアントがセッション情報を保持できるようになっています。

複数のリージョンにデプロイされた ECP
ベースのアプリケーションサーバーとデータベースメンバーには、複数の GCP ゾーンが使用されています。
この構成では、毎秒数千万件のデータベースアクセスと数テラバイトのデータをサポートできます。

大規模な本番構成のサンプル図

図 2.3.1-a のサンプル図は、図 2.3.1-b のリソーステーブルを示します。
含まれているゲートウェイは単なる例であり、組織の標準的なネットワークに合わせて調整できます。

この構成には、フェイルオーバーミラーペア、4 つ以上の ECP
クライアント (アプリケーションサーバー)、およびアプリケーションサーバー当たり 1 つ以上の Web
サーバーが含まれます。フェイルオーバーデータベースミラーのペアは、同一のリージョン内で 2 つの異なる
GCP ゾーンで分割されており、3 番目のゾーンに個別にデプロイされた InterSystems Arbiter と ICM
サーバーでフォールトドメインの保護を確立し、さらなる回復力を備えています。

災害復旧は、前の例と同様に、2 番目の GCP リージョンとゾーンに拡張されています。複数の DR
リージョンは、必要に応じて複数の DR 非同期ミラーメンバーターゲットと使用できます。

以下の GCP VPC プロジェクト内のリソースは、大規模な本番デプロイメントをサポートするための最低推奨事項として推奨されています。GCP リソースは必要に応じて追加または削除することができます。

大規模な本番構成の GCP リソース

???????????????? GCP ??????????????????

InterSystems IRIS シャードクラスタを使用した本番構成

この例では、SQL
を使用したハイブリッドワークロード向けに水平スケーリングされた構成を示しています。InterSystems IRIS
の新しいシャードクラスタ機能が含まれており、複数のシステムをまたぐ SQL
クエリとテーブルの大規模な水平スケーリングを提供しています。InterSystems IRIS
のシャードクラスタとその機能の詳細については、この記事の後半で説明します。

InterSystems IRIS シャードクラスタを使用した本番構成

図 2.4.1-a のサンプル図は、図 2.4.1-b のリソーステーブルを示します。
含まれているゲートウェイは単なる例であり、組織の標準的なネットワークに合わせて調整できます。

この構成には、データノードとして 4 つのミラーペアが含まれます。
それぞれのフェイルオーバーデータベースミラーのペアは、同一のリージョン内で 2 つの異なる GCP
ゾーンで分割されており、3 番目のゾーンに個別にデプロイされた InterSystems Arbiter と ICM
サーバーでフォールトドメインの保護を確立し、さらなる回復力を備えています。

この構成では、クラスタ内のあらゆるデータノードからすべてのデータベースアクセスメソッドを利用することが
できます。大規模な SQL テーブルのデータは、すべてのノード間で物理的に分割されているため、クエリ処理と
データ量の大規模な並列化が実現されます。これらすべての機能を組み合わせることで、単一の InterSystems
IRIS データプラットフォーム内で、新しいデータの取り込みと同時に大規模な分析 SQL
クエリを実行するなど、あらゆる複雑なハイブリッドワークロードをサポートできるようになります。

上の図と、下のテーブルの「リソースタイプ」列にある「Compute
[Engine]」とは、このドキュメントのセクション 3.1
で説明されているとおり、GCP（仮想）サーバーインスタンスを表す GCP 用語です。この記事の後半で説明す
るクラスタアーキテクチャでの「計算ノード」の使用を表したり暗示するものではありません。

以下の GCP VPC
プロジェクト内のリソースは、シャードクラスタをサポートするための最低推奨事項として推奨されています。
GCP リソースは必要に応じて追加または削除することができます。

シャードクラスタを使用した本番構成の GCP リソース

???????????????????? GCP ??????????????????

クラウドの基礎概念

Google Cloud Platform（GCP）は、IaaS（サービスとしてのインフラストラクチャ）向けの機能性豊かなクラウ

ド環境を提供しています。新しい InterSystems IRIS データプラットフォームによるコンテナベースの開発運用など、InterSystems の全製品に完全に対応していますが、あらゆるプラットフォームやデプロイメントモデルと同様に、パフォーマンス、可用性、システム運用、高可用性、災害復旧、セキュリティ制御、およびその他の管理手順などの環境に関わるすべての側面が正しく機能するように注意を払う必要があります。このドキュメントでは、Compute、Storage、および Networking という、すべてのクラウドデプロイメントの 3 つの主要コンポーネントについて説明します。

Compute Engine (仮想マシン)

GCP 内には、仮想 CPU とメモリの仕様と関連するストレージオプションを数多く備えた Compute Engine リソースで利用できるオプションがいくつかあります。GCP 内で注意する項目の 1 つとして、あるマシンタイプの vCPU 数は、1 つの vCPU に相当することです。これはハイパーバイザー層にある物理ホスト上の 1 つのハイパースレッドということになります。

このドキュメントでは、n1-standard* と n1-highmem* インスタンスタイプが使用されており、ほとんどの GCP デプロイメントリージョンで最も広く利用できるインスタンスです。ただし、大量のデータをメモリにキャッシュする非常に大型の作業データセットにおいては、n1-ultramem* インスタンスタイプを使用することが優れたオプションと言えます。特記されている場合を除き、インスタンス可用性ポリシーなどのデフォルトのインスタンス設定やその他の高度な機能が使用されています。さまざまなマシンタイプの詳細は、[こちら](#)を参照してください。

ディスクストレージ

InterSystems 製品に最も直接関係しているストレージタイプは、永続ディスクタイプですが、データ可用性の制限を理解して対応できる場合には、ローカルストレージを使用して高度なパフォーマンスを実現することができます。Cloud Storage (バケット) といったその他のオプションもいくつかありますが、それらは InterSystems IRIS データプラットフォームの運用をサポートするというよりも、個別のアプリケーションの要件に特化したオプションです。

ほかのほとんどのクラウドプロバイダと同様に、GCP でも、各 Compute Engine に関連付けられる永続ストレージの容量に制限があります。これらの制限には、各ディスクの最大サイズ、各 Compute Engine に接続される永続ディスクの数量、永続ディスク当たりの IOPS 量と各 Compute Engine インスタンスの総合的な IOPS 限界などがあります。さらに、ディスク容量 1 GB あたりの IOPS 制限もあるため、希望する IOPS レートを得るには、より多くのディスク容量をプロビジョニングする必要があります。

これらの制限は、時間の経過とともに変化する可能性があるため、適宜、Google に確認する必要があります。

ディスクボリュームの永続ストレージタイプには、標準永続ディスクと SSD 永続ディスクの 2 種類があります。予測可能な低レイテンシ IOPS とより高いスループットを必要とする本番ワークロードには、SSD 永続ディスクがより適しています。標準永続ディスクは、非本番環境の開発とテストやアーカイブタイプのワークロード向けのより経済的なオプションです。

さまざまなディスクタイプと制限の詳細については、[こちら](#)を参照してください。

VPC ネットワーキング

InterSystems IRIS データプラットフォームの多様なコンポーネントのサポートとともに、適切なネットワークセキュリティコントロール、各種ゲートウェイ、ルーティング、内部 IP アドレス割り当て、ネットワークインターフェース分離、およびアクセス制御を提供するには、仮想プライベートクラウド (VPC) ネットワークの使用が強く推奨されます。VPC の例は、このドキュメントに含まれる例で詳しく説明します。

VPC ネットワーキングとファイアウォールの詳細については、[こちら](#)を参照してください。

仮想プライベートクラウド (VPC) の概要

GCP VPC はほかのクラウドプロバイダとは少し異なり、簡素化とより優れた柔軟性を実現しています。概念の比較は、[こちら](#)を参照してください。

GCP プロジェクト内では、プロジェクトごとに複数の VPC を使用でき（現在、プロジェクトあたり最大 5 個）、自動モードとカスタムモードの 2 つのオプションで VPC ネットワークを作成できます。

それぞれの詳細は、[こちら](#)を参照してください。

ほとんどの大規模なクラウドデプロイメントでは、複数の VPC をプロビジョニングして、さまざまなゲートウェイタイプをアプリケーション中心の VPC から分離し、インバウンドとアウトバウンドの通信に VPC ピアリングを活用しています。会社で使用されているサブネットと組織のファイアウォールルールの詳細について、ネットワーク管理者に相談することを強くお勧めします。VPC ピアリングについては、このドキュメントで説明していません。

このドキュメントに含まれる例では、3 つのサブネットを持つ単一の VPC を使用して、予測可能なレイテンシと帯域幅、およびさまざまな InterSystems IRIS コンポーネントのセキュリティ分離を得るために、コンポーネントのネットワーク分離を行っています。

ネットワークゲートウェイとサブネットの定義

このドキュメントでは、インターネットとセキュア VPN 接続をサポートするために、2 つのゲートウェイを使用した例を示しています。アプリケーションに適度なセキュリティを提供するために、各侵入アクセスには、適切なファイアウォールとルーティングのルールが必要です。ルートの使用方法に関する詳細については、[こちら](#)を参照してください。

InterSystems IRIS データプラットフォーム専用のアーキテクチャ例では、3 つのサブネットが使用されています。これらの個別のネットワークサブネットとネットワークインターフェースを使用することで、セキュリティコントロールと帯域幅の保護に柔軟性を持たせ、上記の 3 つの主要コンポーネントをそれぞれ監視することができます。さまざまな使用事例に関する詳細は、[こちら](#)を参照してください。

複数のネットワークインターフェースを備えた仮想マシンインスタンスの作成に関する詳細は、[こちら](#)を参照してください。

これらの例には、次のサブネットが含まれます。

インバウンド接続ユーザーとクエリ用のユーザースペースネットワーク
シャードノード間通信用のシャードネットワーク
各データノード

の同期レプリケーションと自動フェイルオーバーを使用して高可用性を実現するミラーリングネットワーク

注意: フェイルオーバー同期データベースミラーリングは、単一の GCP リージョン内で相互接続のレイテンシが低い複数のゾーンでのみ推奨されます。リージョン間のレイテンシは非常に高いことが通例であるため、特に更新が頻繁に行われるデプロイメントにおいては、良好なユーザーエクスペリエンスを提供することができません。

内部ロードバランサー

ほとんどの IaaS クラウドプロバイダーには、自動データベースフェイルオーバー設計で一般的に使用される仮想 IP (VIP) アドレスに対応できる能力が欠けています。これを解決するために、ミラー対応かつ自動化する VIP 機能に依存しないように、最も一般的に使用されるいくつかの接続方法 (特に ECP クライアントや Web ゲートウェイ) が InterSystems IRIS 内で強化されています。

xDBC、ダイレクト TCP/IP ソケット、またはその他のダイレクトコネクトプロトコルなどの接続方法には VIP のようなアドレスを使用する必要があります。

そういったインバウンドプロトコルをサポートするために、InterSystems のデータベースミラーリング技術では、`mirrorstatus.cwx` というヘルスチェックステータスページを使って、それらの接続方法の自動フェイルオーバーを GCP

内で提供できるようになっています。VIP のようなロードバランサーの機能性を達成するためにロードバランサーと対話し、アクティブなプライマリメンバーにのみトラフィックをダイレクトすることで、完全かつ堅牢な高可用性設計を GCP 内で作り上げています。

ロードバランサーを使用して VIP のような機能を提供する方法の詳細については、[こちら](#)を参照してください。

VPC トポロジーの例

以下の図 4.3-a では、すべてのコンポーネントを組み合わせ、次の特性を持つ VPC のレイアウトを示しています。

- 高可用性を得るために、リージョン内の複数のゾーンを活用する
- 災害復旧を可能にするために、2 つのリージョンを提供する
- ネットワーク分離を実施するために、複数のサブネットを使用する
- インターネットと VPN の両方の接続を得るために、個別のゲートウェイを含める
- ミラーメンバーが IP フェイルオーバーを行えるように、クラウドロードバランサーを使用する

永続ストレージの概要

はじめに説明したとおり、GCP 永続ディスク、特に SSD 永続ディスクタイプの使用をお勧めしています。

読み取りと書き込みの IOPS レートがより高く、トランザクションと分析用のデータベースワークロードに必要なレイテンシが低いため、SSD 永続ディスクが推奨されています。特定の状況でローカル SSD を使用できることもありますが、ローカル SSD のパフォーマンス向上には、可用性、耐久性、および柔軟性のトレードオフが伴うことに注意してください。

ローカル SSD データ永続性の詳細については、[こちら](#)を参照してください。ローカル SSD データが保持される場合とされない場合を理解することができます。

LVM ストライピング

ほかのクラウドプロバイダと同様に、GCP においても、IOPS、容量、および仮想マシンインスタンス当たりのデバイス数に関してストレージに対する制限を課しています。GCP ドキュメンテーションで現在の制限を確認してください。[こちら](#)から参照できます。

これらの制限により、データベースインスタンスの単一ディスクデバイスの IOPS 以上に最大化するために、LVM ストライピングが必要となります。提供されている仮想マシンインスタンスの例では、以下のディスクレイアウトが推奨されています。SSD 永続ディスクに関連するパフォーマンス制限については、[こちら](#)を参照してください。

注意: 現在の仮想マシンインスタンス当たりの最大永続ディスク数は 16 個ですが、GCP では「ベータ版」として 128 個への増加を記載しています。喜ばしい拡張です。

LVM ストライピングのメリットによって、ランダムな IO ワークロードをより多くのデバイスに分散し、ディスクキューを継承することができます。以下は、データベースボリュームグループに対して、Linux で LVM ストライピングを使用する方法の例を示しています。この例では、物理エクステント (PE) サイズが 4 MB の LVM PE ストライプで 4 つのディスクを使用します。または、必要に応じて、より大きな PE サイズを使用することもできます。

- 手順 1: 必要に応じて、標準または SSD 永続ディスクを作成します
- 手順 2: “`lsblk -do NAME,SCHED`” を使用して、各ディスクデバイスの IO スケジューラが NOOP であることを確認します
- 手順 3: “`lsblk -do KNAME,TYPE,SIZE,MODEL`” を使用して、ディスクデバイスを識別します
- 手順 4: 新しいディスクデバイスでボリュームグループを作成します
 - `vgcreate s 4M`
 - **例:** `vgcreate -s 4M vgirisdb /dev/sd[h-k]`
- 手順 4: 論理ボリュームを作成します
 - `lvcreate n -L -i -l 4MB`
 - **例:** `lvcreate -n lvirisdb01 -L 1000G -i 4 -l 4M vgirisdb`
- 手順 5: ファイルシステムを作成します
 - `mkfs.xfs K`
 - **例:** `mkfs.xfs -K /dev/vgirisdb/lvirisdb01`
- 手順 6: ファイルシステムをマウントします
 - 次のマウントエントリで `/etc/fstab` を編集します
 - `/dev/mapper/vgirisdb-lvirisdb01 /vol-iris/db xfs defaults 0 0`

- `mount /vol-iris/db`

上記の表を使用すると、各 InterSystems IRIS サーバーに、SYS 用ディスク 2 個、DB 用ディスク 4 個、プライマリジャーナル用ディスク 2 個、および代替ジャーナル用ディスク 2 個を備えた以下の構成が作られます。

成長に関しては、LVM では中断することなく、必要に応じてデバイスと論理ボリュームを拡張できます。LVM ボリュームの継続的な管理と拡張のベストプラクティスについては、Linux のドキュメンテーションを参照してください。

注意: データベースと書き込みイメージジャーナルファイルの両方で非同期 IO を有効にすることを強くお勧めします。Linux

での有効化に関する詳細については、次のコミュニティ記事を参照してください:

<https://community.intersystems.com/post/lvm-pe-striping-maximize-hyper-converged-storage-throughput>

プロビジョニング

InterSystem IRIS に InterSystems Cloud Manager (ICM) という新しいツールがあります。ICM は、多くのタスクを実行し、InterSystems IRIS データプラットフォームをプロビジョニングするためのオプションを多数提供しています。ICM は Docker イメージとして提供され、堅牢な GCP クラウドベースのソリューションをプロビジョニングするために必要なすべての機能を含んでいます。

ICM は現在、以下のプラットフォームでのプロビジョニングをサポートしています。

- Google Cloud Platform (GCP)
- GovCloud を含む Amazon Web Services (AWS / GovCloud)
- 政府を含む Microsoft Azure Resource Manager (ARM / MAG)
- VMware vSphere (ESXi)

ICM と Docker は、デスクトップ/ノートブックワークステーションから実行することも、小規模な専用の集中型「プロビジョニング」サーバーと集中型レポジトリを持つことも可能です。

アプリケーションのライフサイクルにおける ICM の役割は、定義 -> プロビジョン -> デプロイ -> 管理です。

ICM のインストールと Docker との使用に関する詳細は、[こちら](#)を参照してください。

注意: クラウドデプロイメントでは、ICM を使用する必要はありません。tar-ball ディストリビューションを使用した従来のインストールとデプロイメントは完全にサポートされており、提供されていますが、クラウドデプロイメントでのプロビジョニングと管理を簡単化できる ICM の使用をお勧めします。

コンテナの監視

ICM には、コンテナベースのデプロイメント向けに Weave Scope を使用した基本的な監視機能が含まれています。デフォルトではデプロイされないため、defaults ファイルの

Monitor フィールドを使用して指定する必要があります。

ICM

を使用した監視、オーケストレーション、およびスケジューリングに関する詳細は、[こちら](#)を参照してください。

Weave Scope の概要とドキュメンテーションは、[こちら](#)をご覧ください。

高可用性

InterSystems のデータベースミラーリングは、あらゆるクラウド環境で最も高度な可用性を提供します。直接インスタンスレベルである程度の仮想マシンの回復力を提供するオプションがあります。GCP で利用できる各種ポリシーに関する詳細は、[こちら](#)を参照してください。

前の方のセクションでは、クラウドロードバランサーがデータベースミラーリングを使用して仮想 IP (VIP のような) 機能に自動 IP アドレスフェイルオーバーを提供する方法について説明しました。クラウドロードバランサーは、前の「[内部ロードバランサー](#)」セクションで述べた `mirrorstatus.cmx` というヘルスチェックステータスページを使用します。データベースミラーリングには、自動フェイルオーバー付きの同期ミラーリングと非同期ミラーリングという 2 つのモードがありますが、この例では、同期フェイルオーバーミラーリングについて説明しています。ミラーリングの詳細については、[こちら](#)を参照してください。

最も基本的なミラーリング構成は、アービター制御構成で一組のフェイルオーバーミラーメンバーを使用する構成です。アービターは同一リージョン内の 3 番目のゾーンに配置されており、アービターと片方のミラーメンバーに影響を与える可能性のあるゾーンの停止から保護します。

ネットワーク構成でミラーリングをセットアップする方法はたくさんありますが、この例では、このドキュメントの「[ネットワークゲートウェイとサブネットの定義](#)」セクションで定義したネットワークサブネットを使用します。IP アドレススキームの例は以下のセクションで提供しています。このセクションでは、ネットワークインターフェースと指定されたサブネットについてのみ示しています。

災害復旧

InterSystems のデータベースミラーリングは、高可用性機能を拡張することで、別の GCP 地理リージョンへの災害復旧もサポートし、GCP の全リージョンがオフラインになるという万が一の事態に備え、運営の弾力性をサポートします。アプリケーションがそのような停止にどのようにして耐えるかは、目標復旧時間 (RTO) と目標復旧ポイント (RPO) によって異なります。これらは、適切な災害復旧計画を作成するために必要な分析の初期フレームを提供するものです。以下のリンクでは、アプリケーションの災害復旧計画を作成する際に検討すべき項目のガイドが提供されています。
<https://cloud.google.com/solutions/designing-a-disaster-recovery-plan> および
<https://cloud.google.com/solutions/disaster-recovery-cookbook>

非同期データベースミラーリング

InterSystems IRIS データプラットフォームのデータベースミラーリングは、GCP ゾーンとリージョン間のデータ

レプリケーションを非同期に実行する堅牢な機能を提供しているため、災害復旧計画の RTO と RPO の目標をサポートする上で役立ちます。非同期ミラーメンバーの詳細については、[こちら](#)を参照してください。

前の高可用性のセクションと同様に、クラウドロードバランサーは、自動 IP アドレスフェイルオーバーを仮想 IP (VIP のような) 機能に提供して、前の「内部ロードバランサー」セクションで述べたのと同じ `mirrorstatus.cmx` ヘルスチェックステータスページを使用して DR 非同期ミラーリングも提供することができます。

この例では、DR 非同期フェイルオーバーミラーリングは、InterSystems IRIS デプロイメントが稼働しているゾーンやリージョンに関係なく、上流システムとクライアントワークステーションに単一のエニーキャスト IP アドレスを提供する GCP Global Load Balancing サービスの導入とともにカバーされるようになります。

GCP のメリットの 1 つは、ロードバランサーがソフトウェアによって定義されたグローバルリソースであり、特定のリージョンにバインドされないことです。このため、そしてインスタンスやデバイススペースのソリューションでないため、リージョン全体で単一のサービスを活用できる特有の機能が可能になります。単一のエニーキャスト IP を使用した GCP Global Load Balancing の詳細については、[こちら](#)を参照してください。

上の例では、3 つすべての InterSystems IRIS インスタンスは GCP Global Load Balancer に渡され、ゾーンやリージョンに関係なく、プライマリミラーとして動作しているミラーメンバーにのみトラフィックが転送されるようになります。

シャードクラスタ

InterSystems IRIS には包括的な機能セットが含まれており、ワークロードの性質とワークロードが直面している特定のパフォーマンスの課題に応じて、単独または組み合わせて適用できます。機能セットの 1 つであるシャーディングは、データとその関連するキャッシュの両方を複数のサーバーに分割することで、クエリとデータの取り込みを行うための柔軟で安価なパフォーマンスの拡張を行いながら、リソースを非常に効率的に利用することで、インフラストラクチャの価値を最大化することができます。InterSystems IRIS のシャードクラスタは、広範なアプリケーション、特に以下の 1 つ以上の項目を含むワークロードを使用するアプリケーションに、大きなパフォーマンスのメリットを提供できます。

- 大量または高速データ取り込み、またはその両方。
- 比較的大規模なデータセット、大量のデータを返すクエリ、またはその両方。
- ディスク上の大量のデータをスキャンしたり、重要な計算作業を伴ったりなど、大量のデータ処理を行う複雑なクエリ。

こういった要因は、それ自体でシャーディングから得られる可能性のある利益に変化をもたらしますが、これらが組み合わさった場合はそのメリットがさらに高まる可能性があります。たとえば、大量データの迅速な取り込み、大規模なデータセット、および大量のデータを取得して処理する複雑なクエリという 3 つのすべての要因が組み合わさった場合、今日の分析ワークロードの多くでシャーディングを利用する価値が非常に高くなります。

これらの特性はすべてデータに関係しています。InterSystems IRIS のシャーディングの主な機能は、データボリュームに対して拡張することだからです。ただし、シャードクラスタには、一部またはすべてのデータ関連の要因が伴うワークロードで、多数のユーザーから非常に高いクエリ量が発生する場合に、ユーザーのボリュームに合わせて拡張する機能も含まれます。シャーディングは、垂直スケーリングと組み合わせることもできます。

運用の概要

シャードアーキテクチャの目的は、データとそれに関連するキャッシュを複数のシステム間で分割することにあります。シャードクラスタは、データノードと呼ばれる複数の InterSystems IRIS インスタンス間で、大量のデータベーステーブルを物理的に水平に（行ごとに）分割します。その一方で、アプリケーションが任意のノードを介してこれらのテーブルに透過的にアクセスし、1

つの論理的な結合としてデータセット全体を捉えられるようにします。このアーキテクチャには、次の3つのメリットがあります。

- **????:**

クエリは、データノードで並列に実行され、結果は、アプリケーションが接続されたノードによってマージ、結合され、完全なクエリ結果としてアプリケーションに返されます。多くの場合、実行速度が大幅に改善されます。

- **分割されたキャッシュ:**

単一のインスタンスのキャッシュをデータセット全体で使用するのではなく、各ノードにそれが格納するシャーディングされたテーブルのデータ分割専用の独自キャッシュがあります。そのため、キャッシュのオーバーフローやパフォーマンスを低下するディスク読み取りのリスクが大幅に軽減されます。

- **並列読み込み:**

データをデータノードに並列に読み込めるため、取り込みワークロードとクエリワークロード間のキャッシュとディスクの競合が緩和され、両者のパフォーマンスが改善されます。

InterSystems IRIS のシャードクラスタの詳細については、[こちら](#)を参照してください。

シャーディングの要素とインスタンスタイプ

シャードクラスタは、少なくとも1つのデータノードと、特定のパフォーマンスやワークロード要件に必要な場合は、オプションの数の計算ノードで構成されます。これら2つのノードタイプは単純なビルディングブロックで、シンプルで透過的かつ効果的なスケーリングモデルを提供しています。

データノード

データノードはデータを格納します。物理レベルでは、シャーディングされたテーブル [\[1\]](#) のデータはクラスタ内のすべてのデータノードに分散され、シャーディングされていないテーブルのデータは、最初のデータノードのみに物理的に格納されます。この区別は、ユーザーに透過的です。最初のノードのストレージ消費量はほかのノードに比べてわずかに高いことがあるという唯一の例外がありますが、シャーディングされたテーブルデータは通常、シャーディングされていないテーブルデータをわずかに上回る程度であるため、この差は無視することができます。

シャーディングされたテーブルデータは、必要に応じて、通常は新しいデータノードを追加した後で、クラスタ全体で再調整できます。

この調整により、分散するデータが均等になるようにノード間でデータの「バケツ」が移動されます。

論理レベルでは、シャーディングされていないテーブルデータと

シャーディングされ

たテーブルのすべてのデータの結合はあ

らゆるノードから可視状態である

ため、クライアントは接続しているノードに関係なく、データセット全体を見ることができます。

メタデータとコードも、すべてのデータノードで共有されます。

シャードクラスタの基本的なアーキテクチャ図は、クラスタ全体で均一に見えるデータノードで構成されています

。クライアントアプリケーションは、任意のノードに接続でき、データがローカルであるかのように処理されます。

[1]

便宜上、ドキュメントを通して「シャードニングされたテーブルデータ

」

とい

う用語に

よって、シャード

ニングとしてマークされる、シャード

ニングをサポートするデータモデルの「エクステンツ」データを表しています。

「シャードニングされていないテーブルデータ」と「

シャードニングされていないデータ

」は、シャードニング可能なエクステンツにあり、そのようにマークされていないデータ、またはシャードニングをまだサポートしていないデータモデルのデータを指します。

データノード

低レイテンシが必要となる高度なシナリオでは、潜在的にデータの一定した流入が発生する可能性があるため、クエリを処理するための透過的なキャッシング層を提供するために計算ノードを追加することができます。

計算ノードはデータをキャッシュします。各計算ノードは、対応するシャードニングされたテーブルデータをキャッシュするデータノードに関連付けられています。さらに、そのデータに加え、クエリを満たすために必要に応じてシャードニングされていないテーブルデータもキャッシュします。

計算ノードは物理的にデータを格納せず、クエリ実行をサポートすることを目的としているため、メモリとCPUに重点を置いてストレージを最小限に抑えるなどによって、そのハードウェアプロファイルをニーズに合わせて調整することができます。取り込みは、ドライバ（xDBC、Spark）で直接、または計算ノードで「ベア」アプリケーションコードが実行されるときにシャードニングマネージャコードによって暗黙的にデータノードに転送されます。

シャードクラスタの図説

シャードクラスタのデプロイにはさまざまな組み合わせがあります。

以下の概要図は、最も一般的なデプロイメントモデルを説明しています。これらの図には、ネットワークゲートウェイと詳細は含まれておらず、シャードクラスタコンポーネントのみに焦点が当てられています。

基本的なシャードクラスタ

以下の図は、単一のリージョンと単一のゾーンにデプロイされた 4 つのデータノードを持つ最も単純なシャードクラスタです。クライアント接続をシャードクラスタノードに分散するために、GCP Cloud Load Balancer が使用されています。

この基本モデルでは、単一の仮想マシンとそれに接続された SSD 永続ストレージに GCP が提供するものを超える回復力や高可用性は提供されていません。インバウンドクライアント接続のネットワークセキュリティ分離と、クライアントトラフィックとシャードクラスタ通信間の帯域幅分離の両方を提供するには、2 つのネットワークインターフェイスアダプターを個別に用意することをお勧めします。

高可用性を備えた基本的なシャードクラスタ

以下の図は、単一のリージョンにデプロイされた 4 つのミラーデータノードとゾーン間で分岐した各ノードのミラーを持つ最も単純なシャードクラスタです。クライアント接続をシャードクラスタノードに分散するために、GCP Cloud Load Balancer が使用されています。

高可用性は、リージョン内のセカンダリゾーンで同期的に複製されたミラーを維持する InterSystems のデータベースミラーリングを使用して提供されています。

インバウンドクライアント接続のネットワークセキュリティ分離と、クライアントトラフィック、シャードクラスタ通信、およびノードペア間の同期ミラートラフィック間の帯域幅分離を提供するには、3 つのネットワークインターフェイスアダプターを個別に用意することをお勧めします。

このデプロイメントモデルでは、このドキュメントの前のセクションで説明したミラーアービターも導入されています。

個別の計算ノードを持つシャードクラスタ

以下の図は、大規模なユーザー/クエリ同時実行向けに、個別の計算ノードと 4 つのデータノードを持つシャードクラスタを拡張しています。Cloud Load Balancer サーバープールには、計算ノードのアドレスのみが含まれます。更新とデータの取り込みは、以前と同様にデータノードに直接更新され続け、超低レイテンシパフォーマンスを維持し、リアルタイムデータ取り込みによるクエリ/分析ワークロード間のリソースの干渉と輻輳を回避します。

このモデルでは、計算/クエリと取り込みを個別にスケーリングできるようにリソースの割り当てを微調整することで、計算またはデータをスケーリングするためだけにリソースを不要に無駄にする代わりに、必要なときに「ジャストインタイム」でリソースを最適化し、経済的でありながらも単純なソリューションを維持することができます。

計算ノードは GCP 自動スケールグループ（自動スケーリング）を非常に簡単に使用できるため、負荷の増減に基

づいて、管理されたインスタンスグループへのインスタンスの追加または削除を自動的に実行することができます。自動スケーリングは、負荷が高まるとインスタンスグループにインスタンスを追加し（アップスケーリング）、インスタンスのニーズが低下するとインスタンスを削除（ダウンスケーリング）します。

GCP 自動スケーリングの詳細については、[こちら](#)を参照してください。

自動スケーリングを使用すると、クラウドベースのアプリケーションは、トラフィックの増加を適切に処理し、リソースのニーズが低下するとコストを削減するのに役立ちます。自動スケーリングポリシーを定義するだけで、オートスケーラーは、測定された負荷に基づいて、自動的にスケーリングを実行します。

バックアップ操作

バックアップ操作には、いくつかのオプションがあります。InterSystems IRIS を使用した GCP デプロイメントでは、次の 3 つのオプションを使用できます。

最初の 2

つ

のオプションは、以下で説明するとおり、スナップショットを作成する前にデータベースによるディスクへの書き込みを一時停止し、スナップショットに成功したら更新を再開するというスナップショットタイプの手順を使用しています。

いずれかのスナップショット方式を使用してクリーンなバックアップを作成するには、次のおおまかな手順を実行します。

- データベースの External Freeze API 呼び出しにより、データベースへの書き込みを一時停止する。
- OSとデータディスクのスナップショットを作成する。
- External Thaw API 呼び出しにより、データベースの書き込みを再開する。
- バックアップファシリティはバックアップ場所にアーカイブする。

External Freeze/Thaw API に関する詳細は、[こちら](#)を参照してください。

注意: このドキュメントにはバックアップのサンプルスクリプトは含まれていませんが、InterSystems 開発者コミュニティに掲載される例を定期的に確認してください。 www.community.intersystems.com

3 つ目のオプションは、InterSystems Online のバックアップです。これは、非常にシンプルな使用事例とインターフェイスを備えたより小規模なデプロイメント向けのエントリーレベルのアプローチです。ただし、データベースのサイズが大きくなるにつれ、スナップショットテクノロジーを使った外部バックアップがベストプラクティスとして推奨されます。外部ファイルのバックアップ、より高速な復元時間、エンタープライズ全体のデータビューと管理ツールなどのメリットがあります。

クリーンで一貫したバックアップを確保するために、整合性チェックなどの追加手順を定期的に追加することができます。

どのオプションを使用するかという決定ポイントは、組織の運用要件とポリシーによって異なります。さまざまなオプションをさらに詳しく検討するには、InterSystems にご相談ください。

GCP 永続ディスクスナップショットのバックアップ

バックアップ操作は、GCP gcloud コマンドライン API と InterSystems External Freeze/Thaw API 機能を使用して実行できます。これにより、本来の 24 時間 365 日の運用弾力性とクリーンな定期バックアップの保証が可能になります。GCP 永続ディスクスナップショットの管理と作成および自動化の詳細については、[こちら](#)を参照してください。

論理ボリュームマネージャー (LVM) のスナップショット

別の方法として、市場に出回っている多くのサードパーティ製バックアップツールを使用する場合は、VM そのものにバックアップエージェントを展開し、論理ボリュームマネージャー (LVM) のスナップショットと組み合わせ、ファイルレベルのバックアップを活用することができます。

このモデルには、Windows または Linux ベースの VM をファイルレベルで復元できるというメリットがあります。このソリューションで注意すべき点は、GCP やほとんどの IaaS クラウドプロバイダはテープメディアを提供しないため、すべてのバックアップリポジトリは、短期アーカイブ用のディスクベースであり、長期保管 (LTR) には BLOB またはバケットタイプの低コストストレージを活用できるということです。このモデルを使用する場合は、ディスクベースのバックアップリポジトリを最も効率的に使用できるように、重複除去テクノロジーをサポートするバックアップ製品を使用することを強くお勧めします。

こういったクラウド対応のバックアップ製品には、Commvault、EMC NetWorker、HPE Data Protector、Veritas NetBackup などさまざまな製品があります。

注意: InterSystems では、これらの製品の比較検証や推奨は行っておりません。個々のお客様の責任の下、バックアップ管理ソフトウェアをお選びください。

Online Backup

小さなデプロイメントでは、組み込みの Online Backup ファシリティもオプションとして考えられます。InterSystems のデータベースオンラインバックアップユーティリティは、データベース内のすべてのブロックをキャプチャしてデータベースファイルにデータをバックアップし、出力をシーケンシャルファイルに書き込みます。この、InterSystems 独自のバックアップメカニズムは、本番システムのユーザーにダウンタイムを引き起こさないように設計されています。Online Backup の詳細については、[こちら](#)を参照してください。

GCP では、オンラインバックアップが完了した後、バックアップ出力ファイルとシステムで使用中のほかのすべてのファイルを、その仮想マシンインスタンスの外部にあるほかのストレージの場所にコピーする必要があります。これには、バックアップ/オブジェクトストレージが適しています。

GCP Storage バケットを使用するには、2 つのオプションがあります。

- gcloud スクリプト API を直接使用して、新しく作成したオンラインバックアップ (とほかの非データベース) ファイルをコピーして操作する。詳細は、[こちら](#)を参照してください。
- Cloud Storage バケットはオブジェクトストレージですが、ストレージバケットをファイルシステムとしてマウントし、永続ディスクと同様に使用する。

Cloud Storage FUSE を使って Cloud Storage バケットをマウントする詳細については、[こちら](#)を参照してください。

[#GCP](#) [#IRIS Analytics Architect](#) [#インターシステムズビジネスソリューションとアーキテクチャ](#) [#クラウド](#)
[#コンテナ化](#) [#システム管理](#) [#プラットフォーム](#) [#高可用性](#) [#InterSystems IRIS](#)

ソースURL:<https://jp.community.intersystems.com/post/google-cloud-platform-gcp%E5%90%91%E3%81%91%E3%81%AEiris%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AB%E3%82%A2%E3%83%BC%E3%82%AD%E3%83%86%E3%82%AF%E3%83%81%E3%83%A3>