記事
Toshihiko Minamoto · 2020年9月23日 45m read

Amazon Web Services（AWS）向け InterSystems IRIS サンプルリファレンスアーキテクチャ

Amazon Web Services（AWS）クラウドは、コンピューティングリソース、ストレージオプション、ネットワークなどのインフラストラクチャサービスの幅広いセットをユーティリティとしてオンデマンドかつ秒単位の従量課金制で提供しています。 新しいサービスは、先行投資なしで迅速にプロビジョニングできます。 これにより、大企業、新興企業、中小企業、公営企業の顧客は、変化するビジネス要件に迅速に対応するために必要なビルディングブロックにアクセスすることができます。

更新: 2019年10月15日

以下の概要と詳細は Amazon 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

AWS グローバルインフラストラクチャ

AWS Cloud
のインフラストラクチャは、リージョンとアベイラビリティーゾーン（AZ）を中心に構築されています。
リージョンは、世界中にある複数の AZ が存在する物理的な場所です。 それぞれの AZ
は、別々の施設にある冗長電源、ネットワーク、接続機能を備えた 1
つ以上の異なるデータセンターで構成されています。 これらの AZ は、単一のデータセンターを使用する場合よりも高い可用性、耐障害性、拡張性を持つ本番アプリケーションとデータベースを運用する機能を提供します。

AWS グローバルインフラストラクチャの詳細は、こちらを参照してください。

AWS のセキュリティとコンプライアンス

クラウドのセキュリティはオンプレミスのデータセンターのセキュリティとほぼ同じですが、設備とハードウェアのメンテナンスにコストがかからないという点が異なります。
クラウドでは、物理サーバーやストレージデバイスを管理する必要はありません。 代わりに、ソフトウェアベースのセキュリティツールを使用して、クラウドリソースに出入りする情報のフローを監視および保護します。

AWS クラウドは、責任共有モデルを提供しています。 AWS
はクラウドのセキュリティを管理しますが、クラウドのセキュリティはユーザーが責任を負います。 つまり、実際のデータセンターと同じように、自身が所有するコンテンツ、プラットフォーム、アプリケーション、システム、およびネットワークを保護するために導入することを選択したセキュリティを制御し続けることができます。

AWS クラウドのセキュリティの詳細は、こちらを参照してください。

AWS が顧客に提供する IT インフラストラクチャは、最良のセキュリティプラクティスとさまざまな IT セキュリティ標準に合わせて設計および管理されています。AWS が準拠している保証プログラムの完全なリストは、こちらを参照してください。

AWS クラウドプラットフォーム

AWS ??????????????????????多くの??????????????????? ????????InterSystems IRIS ????????????????? AWS ??????????????? ??

サービスにアクセスするには、AWS マネジメントコンソール、コマンドラインインターフェース、またはソフトウェア開発キット（SDK）を使用できます。

AWS クラウドプラットフォーム	コンポーネント	詳細
	AWS マネジメントコンソール	AWS マネジメントコンソールの詳細は、こちらを参照して
	AWS コマンドラインインターフェース	AWS コマンドラインインターフェース（CLI）の詳細は、こちらを参照し
	AWS ソフトウェア開発キット（SDK）	AWS ソフトウェア開発キット（SDK）の詳細は、こちらを参照してくだ
	AWS コンピューティング	次のようなさまざまなオプション • AmazonElastic Cloud Computing（EC2）の詳細はこちらを参照して • Amazon EC2 Container Service（ECS）の詳細はこちらを参照してくださ • Amazon EC2 Container Registry（ECR）の詳細はこちらを参照してくだ • Amazon Auto Scaling の詳細はこちらを参照して
	AWS ストレージ	次のようなさまざまなオプション • Amazon Elastic Block Store（EBS）の詳細はこを参照してください。 • Amazon Simple Storage Service（S3）の詳細はこを参照してください。 • Amazon Elastic File System（EFS）の詳細はこちらを参照してくださ
	AWS ネットワーキング	次のようなさまざまなオプション • Amazon Virtual Private Cloud（VPC）の詳細はこを参照してください。 • Amazon Elastic IP

		アドレスの詳細はこちら を参照してください。Ama Interface の詳細はこちらを • Amazon の Linux 向け拡張ネッ トワーキングの詳細はこち を参照してください。 • Amazon Elastic Load Balancing（ELB）の 詳細はこちらを参照してく • Amazon Route 53 の詳細はこちらを参照して

InterSystems IRIS サンプルアーキテクチャ

この記事の一部では、アプリケーション固有のデプロイの出発点として、AWS に InterSystems IRIS をデプロイする場合のサンプルを提供しています。デプロイ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、最も小規模なデプロイ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と共にその他の推奨されるシステム運用について説明しています。これらの運用は、組織の標準的なプラクティスとセキュリティポリシーに対応できるように変更してください。

InterSystems では、ユーザー固有のアプリケーションについて、AWS ベースの InterSystems IRIS デプロイ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。これらの小規模な開発、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の例を検討してください。開発作業用の小規模な構成から、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機能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成長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。

小規模な開発の構成

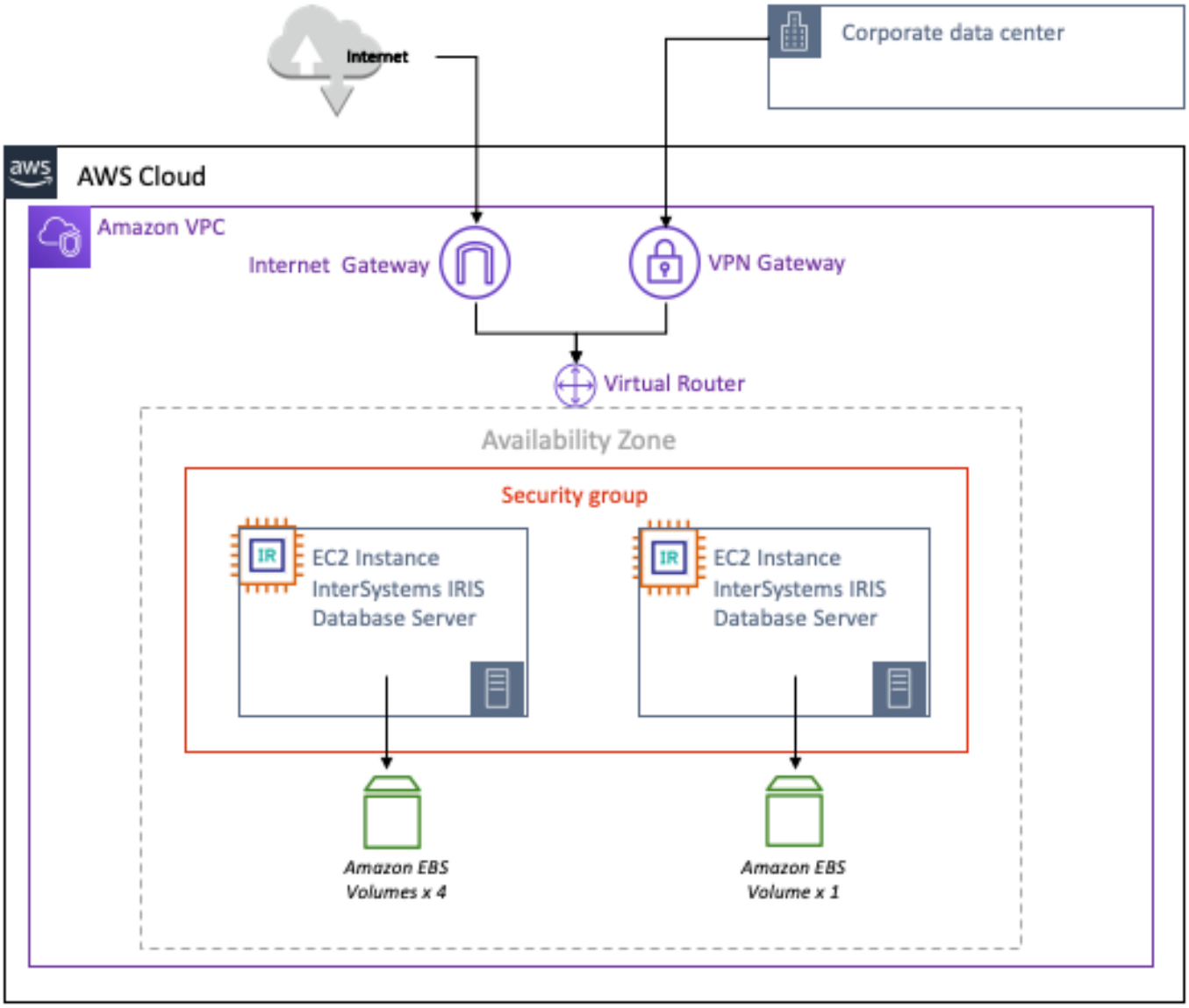
この例では、最小構成を使用して、開発者数最大10 人と 100 GB のデータに対応できる小規模な開発環境を示します。仮想マシンのインスタンスの種類を変更し、EBS ボリュームのストレージを適切に増加するだけで、より多くの開発者と保存データを簡単にサポートできるようになります。

これは開発作業をサポートし、InterSystems IRIS の機能や、必要に応じて 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実践に合わせて調整できます。

図-2.1.1-a: サンプルの小規模開発アーキテクチャ



以下の AWS VPC 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。AWS 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 AWS リソース

以下のテーブルは、小規模構成 AWS リソースのサンプルを示しています。

図 2.1.1-b: 小規模構成 AWS リソースのサンプルテーブル

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。Amazon
は、次のようなネットワークセキュリティのベストプラクティスを提供しています。

https://docs.aws.amazon.com/vpc/index.html#lang/enus

https://docs.aws.amazon.com/quickstart/latest/vpc/architecture.html#best-practices

注意: VM インスタンスが AWS 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。
この実践では懸念を生じてしまう可能性がありますが、AWS
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
インスタンスに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、それを通過するトンネルをセットアップする必要があります。 外部に接する VPC
へのエントリポイントを提供するため、踏み台ホストをプロビジョニングすることができます。

踏み台ホストの使用方法に関する詳細は、こちらを参照してください。

https://aws.amazon.com/blogs/security/controlling-network-access-to-ec2-instances-using-a-bastion-server/

https://docs.aws.amazon.com/quickstart/latest/linux-bastion/architecture.html

本番の構成

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に region-1 内で 2 つのアベイラビリティーゾーンに分割された InterSystems IRIS データベースサーバーの同期ミラーペアと、AWS
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で災害復旧を行うことを目的とした region-2 の 3 番目の DR 非同期ミラーメンバーが含まれます。

マルチ VPC 接続を含む複数リージョンの詳細は、こちらを参照してください。

InterSystems Arbiter と ICM サーバーは、レジリエンシーを高めるために、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、オプションの負荷分散された Web
サーバーのセットも含まれています。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

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ています。ここに記載さ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プラクティスに合わせて調整できます。

図 2.2.1-a: 高可用性と災害復旧を備えたサンプルの本番アーキテクチャ

以下の AWS VPC 内のリソースは、Web
アプリケーションの本番ワークロードをサポートするための最小推奨要件です。AWS
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 AWS リソース

以下の表に、本番構成の AWS リソースのサンプルを示しています。

図 2.2.1-b: 本番構成の AWS リソースの表（続き）

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 の Enterprise Cache Protocol（ECP）を使用したアプリケーションサーバー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。ECP クライアントはデータベースインスタンスのフェイルオーバーが発生した場合でもセッション情報を保持するため、この例ではさらに高いレベルの可用性が実現されています。複数の AWS
アベイラビリティーゾーンが、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ミラーメンバーの両方で使用されています。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。

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 のリソースの表を示しています。ここに記載さ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プラクティス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につき 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 AWS アベイラビリティー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、レジリエンシーを高めています。

災害復旧は、前の例と同様に、2 番目の AWS
リージョンとアベイラビリティー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共に使用できます。

図 2.3.1-a: ECP アプリケーションサーバーを使用したサンプルの大規模本番アーキテクチャ

以下の AWS 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小推奨要件です。AWS
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 AWS リソース

以下の表に、大規模な本番環境構成のサンプル AWS リソースを示しています。

図 2.3.1-b: ECP アプリケーションサーバーを使用した大規模構成の AWS リソースの表

図 2.3.1-b: ECP アプリケーションサーバーを使用した大規模構成の AWS リソースの表（続き）

図 2.3.1-b: ECP アプリケーションサーバーを使用した大規模構成の AWS リソースの表（続き）

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。この構成には 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、この記事の第9章で説明します。

シャードクラスタを使用した本番構成のサンプル図

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。ここに記載さ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プラクティスに合わせて調整できます。

この構成には、4 つのミラーペアがデータノードとして含まれています。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 AWS アベイラビリティー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、レジリエンシーを高めています。

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使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両方を大規模に並列化できます。これらすべての機能を組み合わせることで、大規模な分析 SQL クエリの実行と新しいデータの同時取り込みなどのあらゆる複雑なハイブリッドワークロードを単一の InterSystems IRIS データプラットフォーム内でサポートできるようになります。

図 2.4.1-a：高可用性を備えたシャードクラスタを使用した本番環境構成のサンプル

上の図と下のテーブルの「リソースタイプ」列にある「EC2」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 AWS 仮想サーバーインスタンスを表す AWS の用語です。 第 9 章で説明されてい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 AWS VPC 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小推奨要件です。AWS 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 AWS リソース

以下のテーブルは、シャードクラスタを使用した本番構成の AWS リソースのサンプルを示しています。

図 2.4.1-b: シャードクラスタを使用したサンプル本番環境構成の AWS リソースの表

クラウドの基礎概念

Amazon Web Services（AWS）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。この記事では、Compute、Storage、および Networking という、すべてのクラウドデプロイの 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）

AWS EC2 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。AWS EC2 ではあるマシンタイプの vCPU 数を参照する場合、1 つの vCPU 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に等しいことに注意する必要があります。

このドキュメントでは m5* および r5* EC2 インスタンスタイプが使用されています。これらは、ほとんどの AWS デプロイ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、非常に大きなメモリを備えた x1* のような他の特殊なインスタンスタイプか、NVMe ローカルインスタンスストレージを備えた i3* を使用することが望ましいです。 AWS のサービスレベル契約（SLA）に関する詳細については、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。S3（バケット）や Elastic File Store（EFS）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、AWS 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 AWS 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、EBS gp2（SSD）、EBS st1（HDD）、EBS io1（SSD）の 3 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、標準の EBS gp2 ディスクがより適しています。標準永続ディスクはより経済的なオプションで、非本番環境の開発やテスト、またはアーカイブタイプのワークロードに適しています。

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共に、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、このドキュメント内の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

AWS VPC 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。 勤務先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は説明していません。

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 を使用してさまざまなコンポーネントのネットワークを分離し、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実現しています。

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、各イングレスアクセスには、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。VPC
ルートテーブル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で使用するための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 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。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。

v. 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**
v. シャードノード間通信用の**シャードネットワーク**
v. 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 AWS
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。 この問題を解決するため、ミラー対応と自動化を行う VIP 機能に依存しないよう、InterSystems IRIS 内では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）が強化されています。

xDBC、TCP/IP ソケットによる直接接続などの接続方法や、その他の直接接続プロトコルについては VIP 同様のアドレスを使用する必要があります。このような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.cxw* 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 AWS 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 AWS 内で作り上げています。

AWS Elastic Load Balancer（ELB）の詳細は、こちらを参照してください。

図 4.2-a: 仮想IPアドレスなしの自動フェイルオーバー

ロードバランサーを使用して VIP 同様の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て、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
- VPC ピアリング、インターネット、および VPN 接続用の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

図 4.3-a: VPC ネットワークトポロジの例

永続ストレージの概要

概要で説明したように、AWS Elastic Block Store（EBS）ボリューム、特に EBS gp2 ボリュームタイプの使用をお勧めします。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、EBS gp2 ボリューム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 PE ストライピング

ほかのクラウドプロバイダと同様に、AWS 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。AWS
のドキュメント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。

このような制限があるため、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 を超えて IOPS
を最大化するには、LVM の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: 現在は Linux EC2 インスタンスごとに最大 40 個の EBS ボリュームがありますが、AWS
のリソース能力は頻繁に変更されますので、最新の制限については AWS
のドキュメントを参照するようにしてください。

図 5.1-a: LVM ボリュームグループ割り当ての例

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ています。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、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、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 -I 4MB
  - **例**: lvcreate -n lvirisdb01 -L 1000G -i 4 -I 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。

図 5.1-b: InterSystems IRIS の LVM 構成

LVM では運用を中断することなく、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 のドキュメント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、コミュニティの記事を参照してください。

プロビジョニング

InterSystems IRIS には 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 AWS 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。

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

ICM と Docker は、デスクトップ/ノートパソコンの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」サーバーと集中型リ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: クラウドのデプロイでは、ICM を使用する必要はありません。tar 形式の配布物を使用する従来のインストールとデプロイの方法は完全にサポートされており、利用できます。ただし、クラウドのデプロイでプロビジョニ

ングと管理を容易にするため、ICM の使用をお勧めします。

コンテナの監視

ICM には、コンテナベースの手プロ委に適した 2 つの基本的な監視機能（*Rancher* および *Weave Scope*）が含まれています。 どちらも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。

Rancher の概要とドキュメントは、こちらを参照してください。

Weave Scope の概要とドキュメントは、こちらを参照してください。

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。AWS は単一の EC2 インスタンスに対する可用性を保証していないため、データベースをミラーリングするには、負荷分散や自動スケールグループと組み合わせることもできるデータベース層が必要です。

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.cxw* 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うい 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アベイラビリティーゾーンの停止を防いでいます。

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。

図 7-a: アービターを使用したミラー構成のサンプル

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 AWS
地理的リージョンへの災害復旧もサポートし、AWS 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、運営のリジリエンシーをサポートします。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://aws.amazon.com/disaster-recovery/

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、AWS アベイラビリティー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.cxw* 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、InterSystems IRIS のデプロイが動作しているアベイラビリティー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 DNS アドレスを提供する AWS Route53 DNS
サービスの導入とともに DR 非同期フェイルオーバーミラーリングがカバーされるようになります。

AWS Route53 の詳細は、こちらを参照してください。

図 8.1-a: AWS Route53 でのDR非同期ミラーリングのサンプル

上の例では、InterSystems IRIS インスタンスのフロントエンドである両方のリージョンの Elastic Load Balancer（ELB）の IP アドレスに Route53 が提供されており、アベイラビリティーゾーンやリージョンに関係なく、有効なプライマリミラーであ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。

- 大量または高速なデータの取り込み、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、大規模な計算作業を必要としたりする、大量のデータ処理を行う複雑なクエリ。

このような要因は、それ自体がシャーディングから得られる潜在的なメリットに影響を与えますが、これらを組み合わせた場合はさらにメリットが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められます。
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。

- **並列処理**

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。

図 9.2.1-a: 基本的なシャードクラスタの図

[1] 便宜上、ドキュメントを通して「*シャーディングされたテーブルデータ*」という用語によって、シャーディングとしてマークされる、シャーディングをサポートするデータモデルの「*エクステント*」データを表しています。

「*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ていないデータ*」は、シャーディング可能なエクステントにあり、そのようにマークされていないデータ、またはシャーディ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。

計算ノード

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、対応するシャーディ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ディングされていないテーブルデータもキャッシュします。

図 9.2.2-a: 計算ノードを含むシャードクラスタ

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、ドライバー（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ディングマネージャー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、AWS Elastic Load Balancer（ELB）が使用されています。

図 9.3.1-a: 基本的なシャードクラスタ

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 AWS が提供するものを超えるレジリエンシー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実現するには、2 つのネットワークインターフェー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、AWS 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実現するには、3 つのネットワークインターフェースアダプターを個別に用意することをお勧めします。

図 9.3.2-a: 高可用性を備えた基本的なシャードクラスタ

このデプロイモデルでは、この記事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 AWS 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。

AWS 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。

図 9.3.3-a: 計算ノードとデータノードが分離されたシャードクラスタ

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。
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 AWS デプロイでは、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 を呼び出し、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 を呼び出し、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が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ー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は、組織の運用要件とポリシーによって決まります。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。

AWS Elastic Block Store（EBS）スナップショットのバックアップ

バックアップ操作は、AWS CLI 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、実質的に 24 時間 365 日の運用レジリエンシーとクリーンな定期バックアップを実現できます。AWS EBS 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（LVM）のスナップショットと組み合わせて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、AWS 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。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。

Online Backup

小規模なデプロイでは、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。

AWS 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、バケット/オブジェクトストレージが適しています。

AWS Single Storage Space（S3）バケットを使用するには、2 つのオプションがあります。

- *AWS CLI* 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。
    - 詳細については、こちらを参照してください。
- Elastic File Store（EFS）ボリュームをマウントし、永続ディスクと同様に低コストで使用する。
    - EFS の詳細は、こちらを参照してください。

#AWS #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 #InterSystems IRIS for Health

---

ソースURL:https://jp.community.intersystems.com/post/amazon-web-services%EF%BC%88aws%EF%BC%89%E5%90%91%E3%81%91-intersystems-iris-%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AB%E3%83%AA%E3%83%95%E3%82%A1%E3%83%AC%E3%83%B3%E3%82%B9%E3%82%A2%E3%83%BC%E3%82%AD%E3%83%86%E3%82%AF%E3%83%81%E3%83%A3