

ハイブリッドクラウドで メインフレーム・アプリケーション・ モダナイゼーションを加速

Skyla Loomis

Suman Gopinath

Peter McCaffrey

Tim Brooks

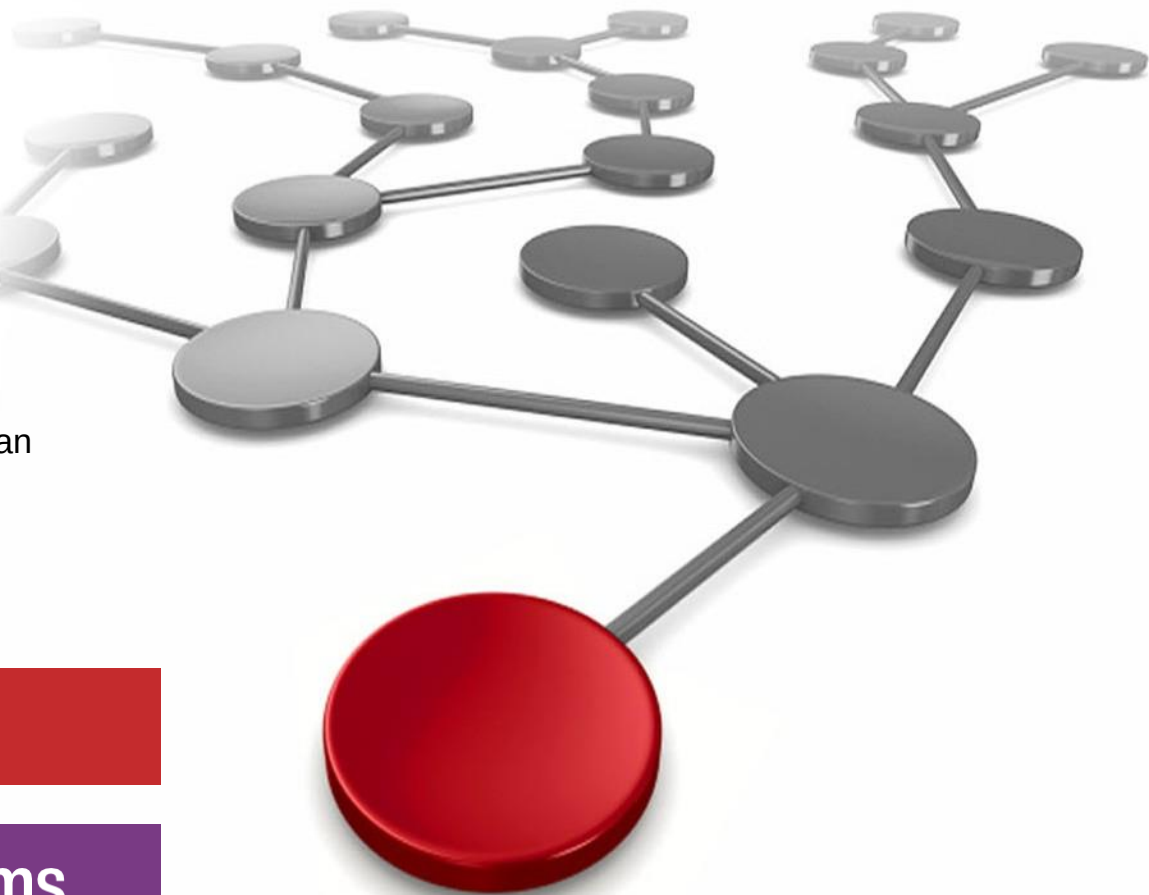
Juergen Holtz

Bryant Panyarachun

Purvi Patel

Mythili Venkatakrishnan

Yichong Yu



 **Cloud**

IBM zSystems

デジタル・トランスフォーメーション、ビジネスの俊敏性、生産性の加速

ハイライト

- ▶ **相乗効果:** IBM zSystems プラットフォームとクラウドの強みを組み合わせることで、より迅速に、より低コストかつ低リスクでメインフレーム・アプリケーションをモダナイズできる。
- ▶ **実証されたベスト・プラクティス:** アプリケーション・モダナイゼーション・ジャーニーを加速させるための処方的アプローチを学習する。
- ▶ **リファレンス・アーキテクチャー:** IBM zSystems、Amazon Web Services (AWS)、Microsoft Azure、IBM Cloud の中から企業に適したソリューションを検討し、導入する。

企業のデジタル・トランスフォーメーションを効果的に推進するための最適な戦略とはどのようなものでしょうか。ビジネスの俊敏性と生産性を高め、サイロを解消し、魅力的な顧客体験を迅速かつ低コストで提供するにはどうすればよいでしょうか。

答え: メインフレーム・アプリケーション・モダナイゼーションを加速させ、ハイブリッドクラウドを活用する。

この IBM Redbooks® では、IBM zSystems® とパブリッククラウド・ソリューションをモダナイゼーション・ジャーニーで併用することで、より迅速に、より低コストかつ低リスクでアプリケーションをモダナイズできるように支援する IBM® の戦略の概要を説明します。¹

また、本書には、企業が望ましい成果を達成するのに役立つ、処方的アプローチ、サポート・テクノロジーのベスト・プラクティス、リファレンス・アーキテクチャー、アクセラレーターが含まれています。

課題

このメインフレーム・アプリケーションとクラウドによる極めて重要な事業変革を実現するためには、次の目標を達成する必要があります。

- ▶ デジタル・トランスフォーメーションの加速
- ▶ 生産性とビジネスの俊敏性の向上
- ▶ スキル・ギャップへの対応

デジタル・トランスフォーメーションの加速

IBM メインフレームとクラウドの両方を活用する企業は、既存の顧客を満足させ、新しい顧客を惹きつけ、競争力を得られるような革新的かつ魅力的な体験を提供するために、新しい方法を構築する必要があります。このような企業は、開発者の生産性を高めることができる、開発者が使い慣れた効果的なツールやプラットフォームを必要としています。

企業のデジタル・トランスフォーメーションが進むにつれて、既存のアプリケーションやデータに対して大きな負荷がかかることがあります。IBM Institute for Business Value の調査では、「5 人の経営幹部のうち 4 人が、競争に遅れないため、メインフレーム・ベースのアプリケーションのモダナイゼーションと、よりオープンなアプローチの採用もふまえ、迅速な変革が必要であると回答しています」。² そうすることで、市場投入までの期間を短縮することができ、競争上の優位性を獲得しようとする際の重要な利点となります。

¹ <https://www.ibm.com/cloud/blog/lower-costs-and-increase-roi-with-application-modernization-on-a-hybrid-cloud>

² <https://www.ibm.com/downloads/cas/7BJPNGND>

生産性とビジネスの俊敏性の向上

企業の最大の資産は人材です。ITを手段として競争力を高めるためには、アプリケーションの開発者と管理者がいつでも適切なテクノロジー（と最も効果的なツール）を使用できるようにして、そうした人材の創造性を引き出し、優れた顧客体験および従業員体験を構築・管理する必要があります。

DevOps 文化を構築することで、生産性と俊敏性に大きな影響を与えることができます。

スキル・ギャップへの対応

IT 業界全体にわたり、スキルの確保は多くの企業にとっての最重要課題です。企業は、メインフレームに特化したスキルを十分に確保できるかどうかを懸念しています。メインフレームのスキルは、組織の他のスキルとはまったく異なる固有のものであるという思い込みも、この懸念に拍車をかけています。例えば、適切な開発者スキルを持たない、メインフレーム上のデータに容易にアクセスできない、古いツールやプロセスを使用しているといった企業は多く、これらすべてが全体的なビジネス・コストを上昇させています。

そこで、IBM zSystems プラットフォームは、人気の高いプログラミング言語、共通の CI/CD ツールチェーンと開発手法、IT 自動化の標準的なアプローチをサポートするようになりました。このプラットフォームを使用することで、企業は組織全体で開発者とサイト・リライアビリティ・エンジニア (SRE) を最適化し、スキル要件に対応できるようになります。

ソリューション

最近の IBM Institute for Business Value の調査「メインフレームのアプリケーション・モダナイゼーション ([Application modernization on the mainframe](#))」では、経営幹部の 71% が、メインフレーム・ベースのアプリケーションがビジネス戦略の中心になっていると回答しています。前述したように、回答者 5 人のうち 4 人が、競争に遅れないため、メインフレーム・ベースのアプリケーションのモダナイゼーションと、クラウドも含むよりオープンなアプローチの採用もふまえ、迅速な変革が必要であると回答しています。

IBM は、お客様の事業変革を後押しするツールやメソドロジーを提供することで、このような課題の解決を支援します。

生産性と俊敏性を向上し、スキル・ギャップを埋めるためには、メインフレーム・アプリケーションをモダナイズして、ハイブリッドクラウドを積極的に活用する必要があります。具体的には、次のベスト・プラクティスを検討してください。

- ▶ ハイブリッドクラウド・アプローチでメインフレーム・アプリケーション・モダナイゼーションを実施する。
- ▶ 継続的なアプリケーション・モダナイゼーション・ジャーニー を推進する。
- ▶ 推奨のクラウド・リファレンス・アーキテクチャーを活用して IBM Cloud®、Amazon Web Services (AWS)、Microsoft Azure を利用する。
- ▶ 実績のある IBM の共創メソドロジー で迅速に開始する。

ハイブリッドクラウドの積極的な活用

アプリケーションのモダナイゼーションとは、企業が現在および将来のニーズを満たすことができるように、アプリケーションを維持、拡張、アクセス、および管理できるように改善するプロセスです。

多くの企業は、従来のオンプレミスから複数のパブリッククラウドや Software as a Service (SaaS) 製品、最近ではエッジ・ベースのコンピューティングまで、さまざまなコンピューティング・モデルをサポートしています。この異種混合モデルの全体的な複雑さは増加の一途をたどっており、開発チームやプロセスがサイロ化することで非効率が発生しています。しかし、すべてを単一のパブリッククラウド・プラットフォームに集約することは、非現実的で、高いコストとリスクが伴います。

IBM はテクノロジーに対して明確で一貫した考えを持っています。ハイブリッドクラウドと人工知能 (AI) は、生産性の向上、より迅速な洞察、より良い意思決定、より優れた顧客体験および従業員体験といった、新時代の到来を促進します。この考えに基づき、IBM の製品やサービスは、お客様がハイブリッドクラウドと AI のビジネス価値を最大限に引き出せるように設計されています。

ハイブリッドクラウドは企業に最も選ばれるアーキテクチャーとなっていますが、それは単一のパブリッククラウドを使用するよりも大きな価値をもたらすからです。この価値は、規模、セキュリティ、使いやすさ、柔軟性、シームレスな体験、イノベーション・サイクルの高速化という形で現れます。さまざまな企業が次々とハイブリッドクラウドを導入し、複数のクラウド、オンプレミス、およびエッジのデータやアプリケーションを統合しています。

メインフレーム (IBM zSystems プラットフォーム) をハイブリッドクラウド戦略の根幹に据えることは、お客様企業でメインフレームの価値を最大限に引き出すための第一歩となります。これは、多くのお客様にとって、メインフレームをハイブリッドクラウド環境の他の部分から分離したり、保護したりしなければならない「孤立したもの」として考える必要がなくなることを意味します。このように考える代わりに、API や IT 自動化でのオープン・スタンダードや、Red Hat OpenShift などのクラウドネイティブなコンテナ・プラットフォームを活用し、メインフレームをハイブリッドクラウドに統合する取り組みを積極的に進めるべきです。

メインフレームを統合することにより、優れたコスト効率で拡張でき、極めて高いレジリエンスとセキュリティを提供する環境はそのままに、ツールやコンプライアンス・プロセスを共有する共通の DevOps 運用モデルを使用することでイノベーションと市場投入までの時間短縮という価値を享受できる、両方の「いいとこ取り」が実現します。

IBM zSystems プラットフォームはセキュアなトランザクション処理に最適なオプション

IBM は、メインフレーム・アプリケーション・モダナイゼーションには、IBM zSystems プラットフォームとクラウドのイノベーションの両方の利点を活用するのが理想的なアプローチであると考えています。

ハイブリッドクラウドでは、既存のアプリケーションとアプリケーションのデプロイメントに最適なプラットフォームを活用することができます (図 1 参照)。

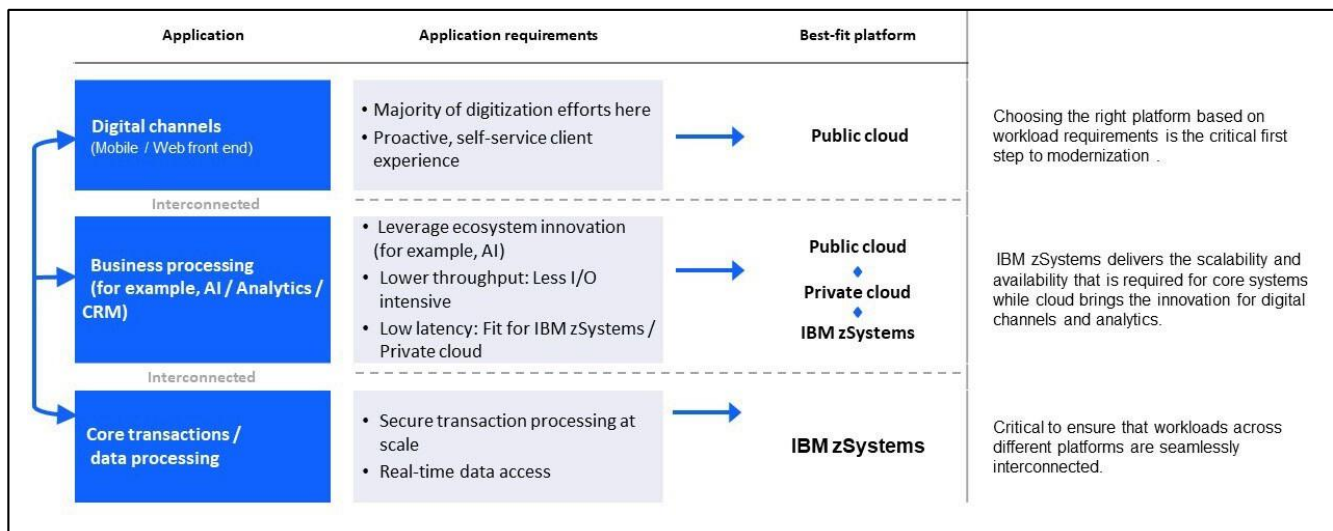


図1 アプリケーションに最適なプラットフォームの選択

IT 業界では、汎用的なインフラストラクチャーに代わって、アプリケーションのワークロードの特性に対応し、効率的に拡張できる特化型インフラストラクチャーが主流になってきています。例えば、ほとんどのパブリッククラウドで、グラフィックスを多用するワークロード、AI や機械学習、ビデオ・トランスコーディングなどに特化した、「ベスト・フィット」なオプションが用意されています。IBM zSystems プラットフォームは、次に挙げる理由から、セキュアなトランザクション処理に対し業界で「ベスト・フィット」であると評価されています。

▶ データ・プライバシー、セキュリティ、およびコンプライアンス

Data in Flight (転送中データ)、Data at Rest (保存されたデータ)、および Data in Use (使用中データ) といった、あらゆる状態のデータを暗号化します。原因が内部であるか外部であるか、偶発的なものか意図的なものかを問わず、データを漏えいから保護します。自動化された証拠収集と報告機能により、コンプライアンスの実証が簡素化されます。ワークロードの変更も必要ありません。

▶ トランザクション処理に必要なダウンタイムはほぼゼロ

データセンターの障害やセキュリティ侵害による個人情報流出が起きると、膨大なコストがかかり、その額は年々増大しています。IBM zSystems プラットフォームを使用すれば、継続的可用性を実現し、データ損失ほぼゼロで復旧し、復旧時間を数日から数秒に短縮することができます。³

▶ トランザクションの整合性と保全性

IBM z16™ は、1 日あたり最大 250 億件の暗号化された IBM z/OS® OLTP トランザクションを処理することができます。アプリケーションの動作の整合性が向上し、データの完全な保全性も確保されます。⁴

▶ スケールとパフォーマンス

IBM zSystems プラットフォームは、パフォーマンスと効率的な拡張性のために最適化されたハードウェアとソフトウェアの垂直統合スタックです。IBM zSystems プラットフォームは、大規模かつ低遅延でトランザクションを処理します。

³ https://www.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/6/897/ENUS222-106/index.html

⁴ <https://www.ibm.com/downloads/cas/MRPZPJPK>

▶ コスト最適化

IBM zSystems プラットフォームの効率的な拡張性を生かして、顧客の要求により迅速に、低コストで対応することができます。Tailored Fit Pricing は、ワークロードのスパイク発生時にオンデマンドでキャパシティを柔軟に拡張して提供することができる、コスト効率の高い使用量ベースの料金体系です。また、IBM Z® Integrated Information Processors (zIIP) を利用することもできます。zIIP は、一部の新しい z/OS アプリケーション専用のプロセッサで、既存アプリケーションのソフトウェア使用料には影響を与えません。zIIP は、IBM zSystems サーバーに新しいアプリケーションを追加するためのコスト効率の高い方法を提供します。

▶ AI による大規模なリアルタイムの洞察

IBM zSystems プラットフォームは、あらゆる顧客のトランザクション・レベルでの対話から得た加速された AI 洞察を駆使し、お客様のビジネス全体で価値創出を支援します。トランザクションに AI を導入することで、トランザクション完了前の不正検知などに役立てることができます。

垂直統合スタック全体にわたる数十年のイノベーションが詰め込まれた IBM zSystems プラットフォームは、ミッション・クリティカルなトランザクション・ワークロードで比類のないスループット、可用性、およびセキュリティを発揮します。セキュアなトランザクション処理を IBM zSystems プラットフォームまたはいくつかのパブリッククラウド上で実行した場合、同じアプリケーションをあるパブリッククラウドで実行した場合に比べて運用コスト (OpEx) を半分にまで削減できるよう設計されています。⁵

デジタル・トランスフォーメーションへの取り組みによってトランザクションの増加が加速し、IT 組織のサービス・レベル・アグリーメント (SLA) を満たす能力が圧迫されています。ハイブリッドクラウド全体にわたりアプリケーションを最適なプラットフォーム上にデプロイすることで、拡張性やパフォーマンスといった非機能要件を達成するための最適なアプローチが実現します。

全体的に、IBM zSystems プラットフォームを実装するハイブリッドクラウド・アプローチは、パブリッククラウドのみのアプローチの **2.5 倍を超える価値**を提供するよう設計されています。⁶

AWS、Microsoft Azure、IBM Cloud との連携を通じたハイブリッド・アプローチによるアプリケーション・モダナイゼーション

IBM zSystems は、AWS、Microsoft Azure、IBM Cloud と連携し、IBM zSystems プラットフォームの中核的な強みと属性を AWS、Microsoft Azure、IBM Cloud で提供される幅広いクラウド・サービスと組み合わせ活用する、ハイブリッド・アプローチによるアプリケーション・モダナイゼーションを推進しています。

IBM zSystems プラットフォームとクラウドを含めたハイブリッド戦略は、アジャイルな DevOps アプローチを活用した迅速な開発を可能にし、メインフレームのアプリケーションとデータへのアクセスを提供し、オープン・ツールと IT 自動化でスキル・ギャップに対応することができます。IBM zSystems プラットフォームを Microsoft Azure、AWS、IBM Cloud と組み合わせることで、メインフレーム・アプリケーション・モダナイゼーションを加速し、俊敏性の向上、コストの最適化、リスクの低減を実現できます。

開始点として、効果的なハイブリッドクラウド・アプローチのためのアプリケーション・モダナイゼーションの取り組みをいくつか特定しました。その中には次のようなものがあります。

▶ デジタル・チャネルからのアクセスの簡素化

クラウド・サービスからのメインフレームのアプリケーションやデータへのアクセスを簡素化します。企業が新しいデジタル・チャネルを通じて顧客体験を強化する際、多くの場合にメインフレームにあるコア・ビジネス・アプリケーションやデータにアクセスする必要性が生じます。

▶ z/OS アプリケーションとクラウド間でのリアルタイムの情報共有

顧客を惹きつけ、ビジネスを差別化できるようなカスタマイズされたオファーを提供することは、ほとんどの企業にとって重要な目標です。この目標を達成する上で、メインフレーム上で動作するコア・ビジネス・アプリケーションと、クラウド上で動作するデジタル・フロントエンド・アプリケーションとの間で、リアルタイムに情報をやり取りすることがますます必要になっています。

⁵ Tailored Fit Pricing for IBM Z Offers Cost Effective Workload Growth Compared to Three Public Cloud Examples, IBM, May 2021

⁶ IBM z16: The platform designed to build the business of tomorrow, May 2022

▶ クラウド上の z/OS アプリケーションでの DevOps 活用

開発者のスピードと俊敏性を高め生産性を向上させるために、エンタープライズ DevOps モデルを活用してハイブリッドクラウド向けの z/OS アプリケーションの開発やモダナイゼーションなどに取り組めます。

▶ IBM zSystems プラットフォームとクラウド全体の IT 自動化

IT 自動化は、これまでも数十年にわたり大規模企業で重要な役割を担ってきましたが、企業が自社のデジタル・イニシアチブをサポートするためにハイブリッドクラウド・アーキテクチャーを採用するようになった今では必要不可欠なものとなっています。

IBM zSystems プラットフォームは、AWS、Microsoft Azure、IBM Cloud と連携し、お客様のニーズや目標を理解し、それらを満たすソリューションを数多く提供できるよう取り組んでいます。

アプリケーション・モダナイゼーション・ジャーニーの開始または推進

IT 環境は絶えず変化しているため、ベスト・プラクティスとして、企業はアプリケーション・モダナイゼーションを一回限りの実装ではなく、継続的なジャーニーと位置付ける必要があります。アプリケーション・モダナイゼーション・ジャーニーとは、既存の投資の可能性を最大化する取り組みです。これを念頭に置くことで、企業は現在および将来のビジネス・ニーズに合わせて進化し成長することができます。継続的なジャーニーには、次のような課題があります。

適切な基盤を構築

アプリケーションの管理とパフォーマンスを合理化しながら、ハードウェアおよびソフトウェアのコストを最適化します。企業は、IBM zSystems プラットフォームをハイブリッドクラウド・ソリューションに統合することで、メインフレーム・アプリケーションへのアクセスを簡素化できます。IT 自動化への標準化されたアプローチを実装することで、企業が一貫性のある生産性の高い方法でハイブリッドクラウド・アーキテクチャーをサポートできるようにします。

さらに、セキュリティとコンプライアンスを確保しながら、アプリケーション・プログラム・インターフェース (API) とデータ・モダナイゼーション戦略を使用して、分析および AI のためのデータ・アクセスを増やすことで、追加の収益の機会を開くことができます。

ビジネスの俊敏性と生産性を向上

エンタープライズ DevOps プロセスと自動化 CI/CD パイプラインに移行します。ビジネス全体でオープンかつ標準を保つ姿勢を維持しながら、このアプローチを柔軟に調整し、現在の優先事項に対応できます。

ジャーニーを加速

実証済みのアプリケーション・モダナイゼーション・パターン、ツール、およびベスト・プラクティスを使用して、価値を生み出すまでの時間を短縮します。これらのリソースには、IBM z/OS、Linux on IBM Z、IBM LinuxONE、およびパブリッククラウドと連携するための「ハウツー」ガイドとショーケースが含まれます。

パブリッククラウドと並行して IBM zSystems を使用してモダナイゼーションへの継続的なアプローチを進めることで、コストを削減し、投資収益率 (ROI) を向上させるための促進剤となる可能性があります。

IBM と共創

クライアント・エンジニアリングと IBM Consulting™ の専門家が、革新的なアイデアを生み出し、そのアイデアを迅速にビジネス価値に変えるためのリソース、テクノロジー、および専門知識をお客様に提供します。IBM の専門家は、お客様とお客様の顧客の問題点に焦点を当てるお手伝いをします。お客様は、自社の IT チームが管理可能なリスクを取りながら最先端のテクノロジーを採用し、製品開発を加速させ、各行動の価値を測定できるように強化することが可能です。着想から構築、拡張に至るまで、成功を導く反復的なフレームワークを使用して、変革の取り組みを主導できます。

エントリー・ポイント

この IBM Redbooks® では、お客様のビジネス・ニーズ、環境、課題、現行プロセスに応じてアプリケーション・モダナイゼーション・ジャーニーを開始または推進できる、いくつかのエントリー・ポイントを説明します。お客様は自社ビジネスに直ちに価値をもたらすエントリー・ポイントを選択すべきです。ベスト・プラクティスとして、次の課題に取り組む必要があります。

- I. 既存のメインフレーム・アプリケーションのコストとパフォーマンスの最適化
- II. アプリケーションの強化とモダナイゼーション
- III. ハイブリッドクラウドにわたる統合
- IV. 情報共有とデータ・アクセスの簡素化
- V. 企業全体の DevOps による俊敏性の向上
- VI. AI 主導の大規模な意思決定
- VII. IT の自動化と標準化

I. 既存のメインフレーム・アプリケーションのコストとパフォーマンスの最適化

IBM は、IBM zSystems プラットフォーム上でアプリケーションを稼働・デプロイする際のコストとパフォーマンスを最適化すべく、メインフレームのハードウェアとソフトウェアに継続的に投資を行ってきました。

このエントリー・ポイントでは、[Tailored Fit Pricing](#) プログラムについて説明します。このプログラムはほとんどのメインフレーム環境に適していますが、とりわけモダナイズされたメインフレーム環境に効果的な商用モデルです。また、このセクションでは、運用コストの削減、既存の IBM zSystems アプリケーションの効率化、およびレジリエンスの向上のために利用できる、さまざまな最適化や機能についても説明します。

Tailored Fit Pricing の活用

Tailored Fit Pricing は、クラウドと同様に、ソフトウェアとハードウェアの使用量に基づいた料金体系オプションを、オンプレミス・コンピューティングに適用することで、従来の料金体系を劇的に簡素化する柔軟な料金体系モデルです。Tailored Fit Pricing for Software では、従来のピークを基準にした料金体系ではなく、処理量、つまり使用量に基づいて料金が決定されます。

この使用量ベース・モデルは、頻繁にスパイクが発生し、不安定で予測不可能な性質を持つ、モダナイズされたワークロードや環境により適していると言えます。IBM zSystems のテクノロジーは、このような性質を持つワークロードを処理する能力が非常に高く、Tailored Fit Pricing も同様に、このようなワークロードに適した料金体系です。処理のスパイクやピークがソフトウェア料金体系における大きな課題であったのはもはや過去のことです。今日の IT 時代の処理に適したテクノロジーと料金体系モデルがあることがわかっていて、今、どのお客様も自信を持ってメインフレーム・アプリケーションをモダナイズできます。

使用量ベースの料金体系を中核とする Tailored Fit Pricing for Software を使用すると、IBM zSystems プラットフォーム上で既存のアプリケーションを管理し、新しいアプリケーションを導入する際に料金体系の予測可能性と透明性を向上させることができます。このモデルでは、上限を設定する必要がないため、ピーク時の料金という制限要因なしに、必要なときに必要なだけマシン・リソースを使用することができます。このモデルは、IT 部門がアプリケーション・モダナイゼーション・ジャーニーを進める中で、コスト分析や、ソフトウェアに対する請求額に見合う効果が得られるかという懸念に基づいてテクノロジーを選択するのではなく、ビジネス成果を意識して選択した「ベスト・フィット」なテクノロジーを基盤にソリューションを設計するよう促します。

Tailored Fit Pricing for Software と Tailored Fit Pricing for Hardware を併用することもできます。後者は、Tailored Fit Pricing プログラムの適合性を、モダナイズされたアプリケーション・ワークロードや環境に拡張します。

企業がすでに永続的に所有しているものに加えて、追加のハードウェアをプロビジョニングして使用することができます。このモデルは、予期しない短い要求のスパイクが起きることがあるお客様において、「常にオン」の追加のハードウェアを必要に応じていつでも使用できる状態にする、というシナリオに適しています。追加のハードウェアはサブスクリプション・ベースでプロビジョニングされ、スパイク発生時にこの追加のハードウェアが使用される度に、細かい粒度で使用料金が請求されます。

図 2 は、Tailored Fit Pricing の例を示しています。

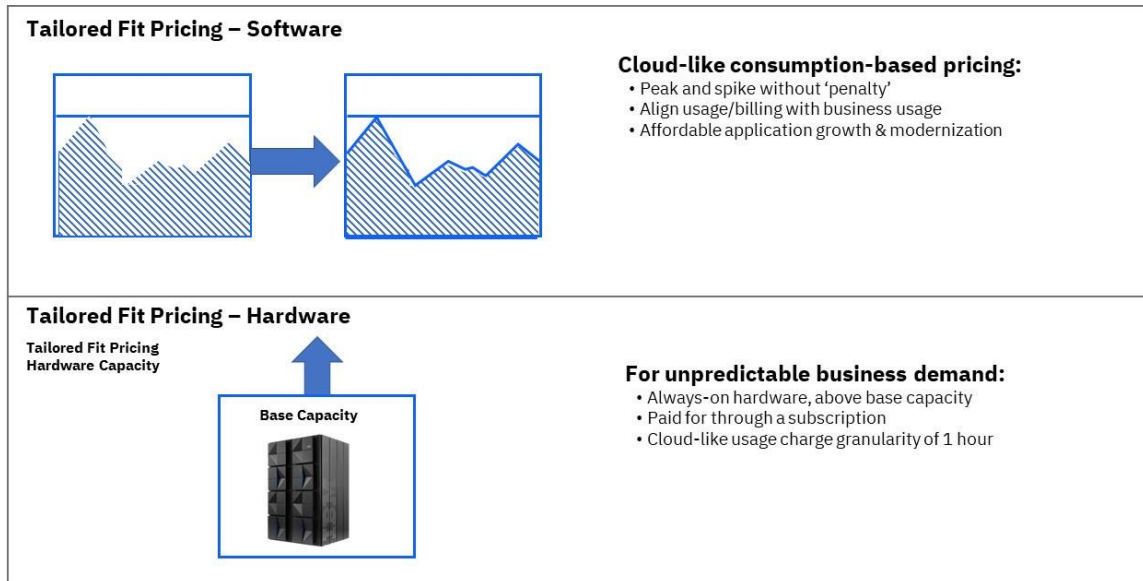


図 2 Tailored Fit Pricing

スペシャリティー・プロセッサの使用

IBM zSystems プラットフォームは、次の 2 つのスペシャリティー・プロセッサを提供します。これらのプロセッサは、セキュリティ、可用性、拡張性に優れた IBM zSystems プラットフォームを活用しながら、企業がアプリケーションの稼働コストを削減できるように設計されています。

▶ IBM zSystems Integrated Information Processor (zIIP)

IBM zIIP は、一部の新しい z/OS アプリケーション専用のプロセッサで、既存アプリケーションのソフトウェア使用料には影響を与えません。IBM zIIP は、IBM zSystems プラットフォームに新しいアプリケーションを追加するためのコスト効率の高い方法を提供します。IBM zIIP は、一部のデータ（データベース）、クラウドやトランザクション（Java）処理のワークロード向けに設計されています。IBM zIIP を利用することで、ユーザーは、Millions of Service Units (MSU) レーティングやマシン・モデルの指定に影響を与えることなく、新しいアプリケーションに追加の処理能力を使用することができます。

zIIP プロセッサを活用できるアプリケーションの種類には、Java アプリケーションおよび z/OS Container Extension (Red Hat OpenShift、データ仮想化、機械学習、API、System XML 構文解析、ONNX モデル、IBM Z AI Data Embedding Library、System Recovery Boost、各種 IBM 製品およびそれらの製品で定義されている ISV 製品)、ならびに IBM Db2® Distributed Relational Database Architecture があります。同じアプリケーションを同等の x86 サーバーやパブリッククラウド・インフラストラクチャーで実行する場合と比較して、潜在的な総所有コスト (TCO) を大幅に節約することができます。

▶ IBM Integrated Facility for Linux (IFL)

IBM IFL は、IBM Z および IBM LinuxONE 上の Linux アプリケーション・ワークロード専用のプロセッサです。IBM zSystems プラットフォームおよび LinuxONE (RedHat を含む) 用の人気の高い Linux オペレーティング・システム、IBM z/VM®、および Kernel-based Virtual Machine (KVM) でサポートされています。IBM IFL は、Linux アプリケーションが IBM zSystems と LinuxONE の大容量の処理能力とオンチップ圧縮アクセラレーションを活用できるようにし、ハイブリッドクラウド・コンピューティングの基盤を提供し、セキュリティとデータ・プライバシー、およびサイバー・レジリエンシーにおける業界最先端の機能を提供します。

IFL を使用するメリットとして、お客様は Linux アプリケーション・インスタンスを統合することで、運用の労力、ソフトウェア・ライセンス料、およびエネルギー消費量や床面積の削減によりコスト削減を実現できます。これは企業の IT 組織のサステナビリティ目標の達成にも貢献します。IBM z16 では、最新のハードウェアの改良機能やスペシャリティー・エンジンをアプリケーションに取り入れることができ、同じ計算集約型アプリケーションを前世代の IBM zSystems で実行する場合と比べてスループットを向上させたり、アプリケーションをミッション・クリティカルなコアに近い場所に置きながら既存の分散システムのコストと複雑さを軽減したりすることができます。

プラットフォームの改良機能の活用

ソフトウェアとハードウェアを最新化し、新しい IBM zSystems プラットフォームのイノベーションを活用することで、現在のメインフレーム環境のコストとパフォーマンスを最適化することができます。IBM z16 の拡張機能を活用することで、企業はトランザクション処理の大幅な改善を実現し、システム全体の効率化をもたらす、より新しい機能の恩恵を受けることができます。IBM z16 では、最新のハードウェアとソフトウェアの拡張機能をアプリケーションに取り入れ、同じ計算集約型アプリケーションを前世代の IBM zSystems で実行する場合と比較して、コストを削減し、スループットを向上させることができます。

例えば、IBM z16 を使用して計算集約型アプリケーションを最新のソフトウェアでコンパイルすると、IBM z14® で同じ計算集約型アプリケーションを稼働する場合と比較して CPU 使用率を削減できます。⁷

IBM z/OS アプリケーションの大部分は、数十年にわたって開発された数百万行に及ぶ COBOL で書かれており、コアなビジネス・プロセスや組織に蓄積された知識が詰め込まれています。COBOL と最新ハードウェアを同時にアップグレードすることで、大幅なパフォーマンスの向上が期待できます。潜在的なメリットの例としては、IBM z16 上で IBM Enterprise COBOL for z/OS 6.4 を使用する場合、Enterprise COBOL 5.2 で構築された計算集約型アプリケーションの CPU 使用量を平均で約 30% 削減できます。

IBM z16 上で [IBM Automatic Binary Optimizer for z/OS 2.2](#) を使用して計算集約型アプリケーションを最適化すると、IBM z15® で IBM Enterprise COBOL for z/OS 4.2 を使用して構築した同じアプリケーションを稼働した場合と比較して、CPU 使用量を平均 60% 削減できます。

IBM は、IBM zSystems アーキテクチャーを使用し、リスクを軽減するためのツールを提供して、モダナイゼーションや Java などの他のプログラミング言語との相互運用性をサポートする最新のプログラミング技術を実現するという目的のもと、COBOL の最適化・拡張に継続的に取り組んでいます。例えば、IBM は次のことを実現しています。

- ▶ 簡素化された COBOL/Java 相互運用機能により、オブジェクト指向の COBOL の書き出しの必要なく、COBOL プログラムの機能を Java で拡張できるようになり、必要な手動の JNI 呼び出し回数が削減されます。
- ▶ AMODE 31 (31 ビット) と AMODE 64 (64 ビット) の COBOL プログラム間の組み込みの相互運用性により、すべてのアプリケーションを 64 ビットに変換することなく、増大する COBOL プログラム・データを処理することができます。そのため、31 ビット・アプリケーションを少しずつ 64 ビットに変換することができます。
- ▶ ユーザー定義関数をサポートし、独自の関数を書き込んで、組み込み関数のように呼び出せるようにして、コードのモジュール性と保守性を向上します。COBOL でユーザー定義関数をサポートするようになったことで、新しい COBOL プログラマーに馴染みのある構造を提供します。
- ▶ IBM Automatic Binary Optimizer for z/OS との統合が改善されたことで、現在 COBOL 6.4 を使用してコンパイルしているモジュールが、再コンパイルすることなく、将来の IBM zSystems ハードウェアの機能強化を活用できます。IBM Automatic Binary Optimizer for z/OS は、コードの変更や再コンパイルなしに、重要なビジネスの COBOL アプリケーションの運用コスト、CPU 使用量、処理時間を削減するように設計されています。

また、あるリリースのミドルウェアから最新のリリースに移行する際にもパフォーマンスの向上が期待できます。管理されたラボ環境で行われた IBM 社内テストの結果、IBM z16™ 上で稼働する IMS 15.3 では、IBM z15 と比較して、ワークロードによって 4%~21% の内部トランザクション率 (ITR) の向上が実証されました。詳しくは、ホワイト・ペーパー「[IBM z16 上での IMS パフォーマンス評価 \(IMS Performance Evaluation on IBM z16\)](#)」を参照してください。

⁷ <https://www.ibm.com/products/cobol-compiler-zos/details>

アプリケーション変更時におけるビジネス・レジリエンスの維持

アプリケーションのワークロードが増加するにつれ、お客様はシステムのレジリエンスを確保するという課題に直面することがあります。負荷しきい値を超過したことによってワークロードが停止するリスクを低減するためには、本番同様のセットアップで負荷テストとスケール・テストを行うことが重要です。コスト面から見て、永続的に余分なキャパシティを購入することは、ほとんどの場合でコスト効果が高いとは言えません。IBM Z Business Resiliency Stress Test (zBuRST) ソリューションを使用することで、企業が実動環境に変更を加える前に、予備の IBM zSystems 物理リソースを使用して実動規模での負荷テストとストレス・テストを行うことができます。希望するプライス・ポイントで実動規模でのテストを行い、ビジネス・レジリエンスを高めることができます。

事業継続性計画の一環として必要不可欠なのが、社内/社外や計画的/計画外のビジネスへの要求と脅威に対して、事業継続を確保しながら対応・適応する能力です。IBM は 99.99999% の可用性にコミットし、投資しています。⁸ このことから、IBM Z Flexible Capacity for Cyber Resiliency ソリューションが導入されました。このオファリングにより、災害復旧時に、異なるサイトの参加している IBM z16 システム間でキャパシティをシフトできるようになります。ターゲット・サイトのターゲット構成は、最大 12 カ月間使用することができます。キャパシティのシフトは、IBM GDPS® を使用することで完全に自動化できます。高い柔軟性が実現し、キャパシティを 2 倍購入する場合と比較して、コストを最適化することができます。規制対象の業界では、このような柔軟性が必須条件であることも少なくありません。詳しくは、「IBM Z Flexible Capacity for Cyber Resiliency」を参照してください。

ビジネス・プラクティスの合理化によるモダナイゼーション

昨今では、管理されたチャージバック・システムを通じて、固定費と実際の使用量に応じた変動費を振り分ける企業が多くなっています。

メインフレーム・アプリケーションのモダナイゼーションによって生産性が向上し、コストが最適化されるのに伴い、企業は効率やコストを内部顧客に反映させる方法を適応させる必要があります。企業は、自社の IT 組織を通じて、このようなコスト削減を資本的支出と運用支出の両面から実現することができます。ほとんどの企業がチャージバックやショーバックによって IT コストを各事業部門に配分できるため、モダナイズ・効率化された IT リソースをより多くの事業部門で共有することができ、部門ごとの運用予算でもコスト削減が見込めます。

II. アプリケーションの強化とモダナイゼーション

企業は、市場の需要に応え、ビジネスの成長を可能にする方法で、アプリケーションを保守、拡張、導入、管理する必要があります。Java アプリケーションと COBOL アプリケーションの対話、既存アプリケーション向けのオープン API の作成、データ仮想化を通じたデータ・ソースへの接続が実現したことで、メインフレーム・アプリケーションを拡張して、新しいビジネス要件をサポートできるようになりました。

パブリッククラウドや Linux on IBM Z 上で稼働するクラウドネイティブ・アプリケーションは、データ処理の種類や遅延の要件に応じて、メインフレームのアプリケーションやデータにアクセスできます。

今日のアプリケーションは、メインフレーム、分散型、パブリックまたはプライベートクラウドなど、複数の IT プラットフォームにまたがって実行されています。多くのコア・ビジネス・アプリケーションは、何十年も前からメインフレームで稼働しています。ビジネス・ニーズの高まりに伴い、アプリケーションは市場の要求に応じて新しい機能を次々に生み出すことが求められています。それらの機能拡張をどのように作成し、アプリケーションをどこにデプロイするかは、アプリケーションの種類や、サービス品質、セキュリティ、データへのアクセス性、および保守性などの非機能要件によって異なります。

このセクションでは、継続的なアプリケーション・モダナイゼーション・ジャーニーの次のエントリー・ポイント、新しい機能を開発するためのベスト・プラクティス、およびそれらをデプロイするのに最適な場所について説明します。

⁸ <https://www.ibm.com/z/resiliency>

新しい開発方法

企業が新しい機能を開発する方法の一つは、既存のアプリケーションやデータの価値を活かして、それらを変更することなくデジタルでの使用向けに提供することです。IBM zSystems プラットフォームは、IBM zSystems プラットフォーム上のコア・アプリケーションから標準的な OpenAPI 仕様に基づく API を開発するためのツールや、API を公開するための堅固で包括的なランタイム環境をサポートします。詳しくは、15 ページの『III. ハイブリッドクラウド全体にわたる統合』を参照してください。

IBM zSystems のサービスやデータを提供する API は、既存のアプリケーションから抽出されることがほとんどですが、API を既存のビジネス・アプリケーションにマッピングし、拡張機能をコーディングすることが必要なケースもあります。より多くの企業が API エコノミーに参加することで、まったく新たな機能の開発が実現します。

数十年の時を経てメインフレームの IT システムがモノリス化し、事業領域がプログラムやトランザクションの範囲によってカバーされなくなっていることもあるでしょう。その場合には DevOps プロセス、ツール、オンデマンド・テスト環境、テスト自動化を導入することで、このような密結合アプリケーションを拡張することができます。アーキテクチャーが理由で効率的な粒度の API を作成できない場合、アーキテクチャーを個別の機能に分解することが必要になります。そうすれば、数十年かけて微調整されてきたロジックやルールを、開発に活用できます。

大きなリスクとコストが伴う「ゼロからの作り直し」ではなく、ビジネス・ニーズの発生に応じて既存のアプリケーションを拡張・強化することがベスト・プラクティスです。そうすることで、アプリケーションのアクティブな部分や変更の可能性が高い部分だけをアップデートすることができます。

IBM Application Discovery and Delivery Intelligence (ADDI) のような自動検出ツールを使用することで、必要に応じて複数の機能に分解できる既存の機能を検出するプロセスをスピードアップすることができます。

新しい機能を開発するこれらのどの方法でも、使用するテクノロジーを自由に選択できるようになりました。IBM zSystems プラットフォームは混合言語サポートを提供します。つまり、新しい機能を開発する際に、同じプログラミング言語を使用してコードを作成することも、開発チームのニーズや機能の特性に適合する別の言語を使用してコードを作成することもできるのです。

適切なコンテキストでの適切なプログラミング言語の使用

機能拡張や新機能に使用できるプログラミング言語は数種類あります。新しいビジネス機能は、現在の z/OS のネイティブ言語である COBOL や PL/I で書くことができます。アプリケーションを自動化されたエンタープライズ CI/CD プロセスに組み込むことで、こうした変更を効率的に管理し、デプロイすることができます。これらの言語の最新のコンパイラと IBM Automatic Binary Optimizer for z/OS for COBOL は、最新のハードウェア機能を使用して、計算の高速化、特に金融アプリケーションで広く使われている 10 進数型の計算を高速化します。

Java のスキルを持つ開発者は市場で非常に見つけやすく、このスキルは IBM zSystems プラットフォーム上のアプリケーションを作成するためにも頻繁に使用されます。また、Java は zIIP へのオフロードが可能であるため、コストの最適化にも有効です。Java で新しい機能を書くことで、市場の豊富な開発者スキルを活用できるようになり、その後徐々にアプリケーションの重要な部分や関連する部分を Java で構築することができます。

IBM の Atruvia 社の [ケース・スタディー](#)によると、Atruvia 社のシニア・エンタープライズ・アーキテクトである Pascal Meyer 氏は、「IBM zSystems での Java の使用は、お客様の競争優位性を高めるためのキー・テクノロジーであると考えています」と述べています。さらに、このケース・スタディーには次のような記述があります。

「IBM zSystems で COBOL に加えて Java を使用することで、開発者は大規模なプログラム書き換えを行うことなく、COBOL のサブルーチンを Java に置き換え、シームレスでリスクの少ない方法で銀行のコア機能を強化することができます。また、分散環境内のソフトウェア・アーキテクトが、より簡単に IMS アプリケーションから直接コア・トランザクション・サービスを呼び出せるようになります」

新しいビジネス機能は、現在の z/OS のネイティブ言語である COBOL や PL/I で書くことができます。アプリケーションを自動化されたエンタープライズ CI/CD プロセスに組み込むことで、こうした変更を効率的に管理し、デプロイすることができます。これらの言語の最新のコンパイラと IBM Automatic Binary Optimizer for z/OS for COBOL は、最新のハードウェア機能を使用して、計算の高速化、特に金融アプリケーションで広く使われている 10 進数型の計算を高速化します。

z/OS のプログラミング言語の中でも、新しい機能を書くのによく使われるのが Java です。同じ作業単位で既存の z/OS リソースを呼び出す必要があるアプリケーションは、多くの場合、IBM CICS® または IMS で Java アプリケーションとして記述・拡張されます。これらのトランザクション・サーバーは、Java インターフェースを使用して簡単にネイティブ・リソースを呼び出したり、JDBC/JMS 仕様に基づいて Db2、IMS データベース、**IBM MQ** などのデータ・ソースを呼び出しするためのライブラリーを提供します。新機能と既存アプリケーションを同じグローバル・トランザクション・スコープで実行できるため、言語の壁を越えてデータベースのコミットにおける保全性を確保することができます。

また、Java で書かれた新しい機能は、既存の COBOL アプリケーションと相互運用することもできます。COBOL-Java の相互運用性は同じトランザクション・コンテキストで管理され、31 ビットと 64 ビットのアドレス・スペースを処理することができます。

図 3 は、IMS におけるこの相互運用性の例を示しています。IMS 依存領域は、COBOL を実行するための永続 31 ビット・アドレス・スペースと、64 ビット JVM で Java を実行するための 64 ビット・アドレス・スペースを作成します。

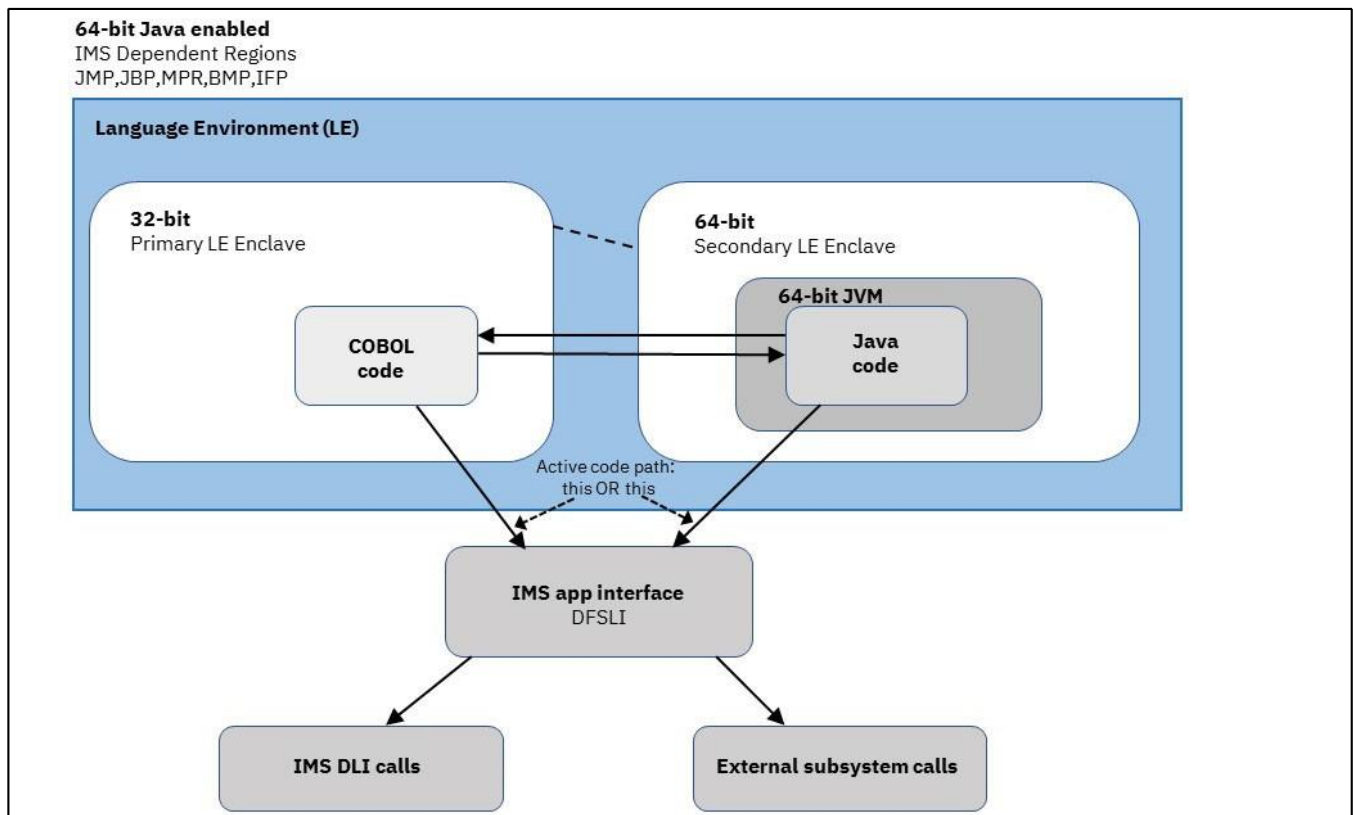


図 3 Java と COBOL の相互運用性

利用可能なスキル・セットに応じて、Node.js、Go、Python などの他の言語を使用して新しいビジネス機能を構築・拡張することができます。開発者は、自身のニーズに合った言語を選択することができるほか、パッケージ・エコシステムを利用してデリバリーを迅速化できます。すべてのユーザー・インターフェースを、Java や最新の言語に移行することをお勧めします。他のアプリケーションやサービスとのゆるい結合を必要とする、粒度の細かい再利用可能なサービスは、多くの場合、クラウドネイティブ・アプリケーションとして記述されます。

13 ページの図 4 は、CICS と IMS で Java 仮想マシン (JVM) と Liberty サーバーを使用する例を示しています。

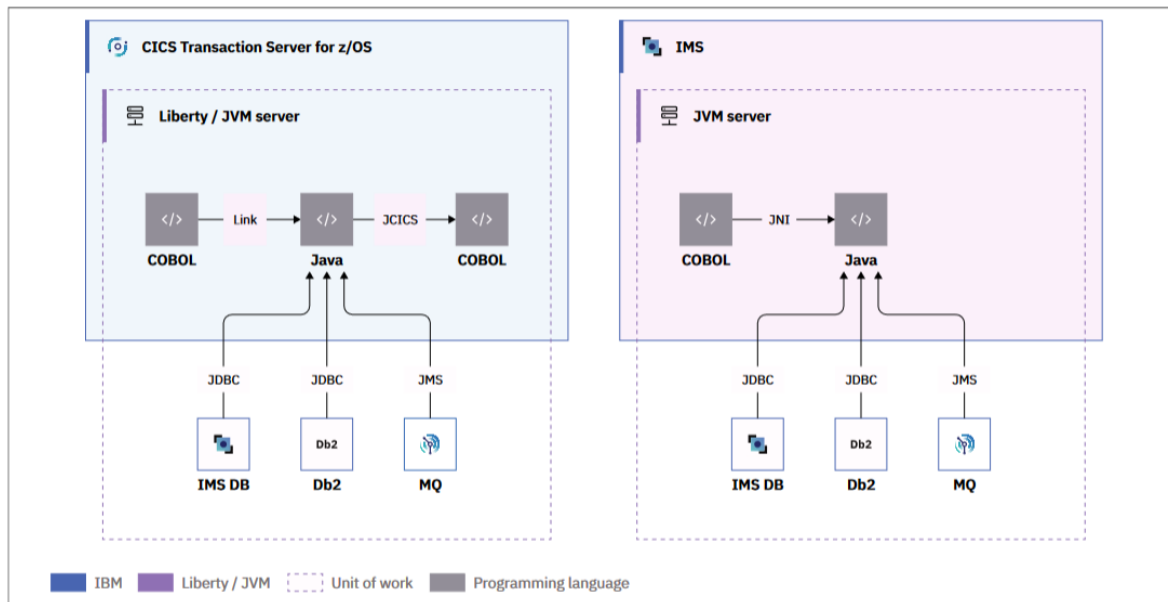


図4 CICS と IMS での JVM または Liberty の使用

アプリケーションの分析

z/OS 上で動作するさまざまなトランザクション・アプリケーションやバッチ・アプリケーションは、数十年前に設計されたものが多く、暗黙知が失われていることがあります。しかしながら、継続的なビジネス・ニーズに対応するための進化は今も続いています。

既存の資産を API 化して新しいビジネス機能を提供する場合でも、既存の資産を拡張して、または IBM zSystems プラットフォーム上のデータ・ストアを利用して新しい機能を開発する場合でも、アプリケーションの理解と発見が重要な役割を果たします。IBM Application Discovery and Delivery Intelligence (ADDI) は、静的分析を基に既存アプリケーションのメタデータ・リポジトリを構築します。これにより、開発チームが API として公開するのに適した資産を抽出したり、データ・リネージュや変更による既存アプリケーションへの影響を自動で調査したりすることが可能になります。

例えば、IBM の Sun Life 社の[ケース・スタディー](#)によると、Sun Life 社は、アプリケーション・コードの検索時間を手動のアプローチと比較して 50% 削減しました。

「開発者は、コードのつながりを特定することに加え、変更が COBOL のルール・ベースの制限に従っているかどうかを知る必要がありました。ADDI を使用することで、分析時間を 35% 削減することができました」

Sun Life 社は、次のようにも述べています。

「これまで、ジョブ・グラフを生成するスキルを持っている開発者は当社に一人もいませんでした」と、Gary Lesage 氏は言います。「ADDI にスケジュール情報を取り込むだけでグラフを生成できる機能は、これまでにない画期的なものでした。チームの全体的な効率向上につながりました」

遅延の低減: クラウドネイティブ・アプリケーションと z/OS アプリケーションのコロケーション

新しいアプリケーションのデプロイ方法は複数あります。マルチクラウド環境にデプロイ可能なスケーラブルなアプリケーションを構築したい場合、Kubernetes 上でクラウドネイティブ・アプリケーションを稼働する方法が多く使用されます。IBM zSystems プラットフォームは、クラウドネイティブ・アプリケーション、従来の z/OS ワークロード、および分散アプリケーションのためのランタイム環境を提供します。

遅延の程度と相互運用性が、環境を決定する際の重要な要因となります。オフ・プラットフォーム (分散型やパブリッククラウド) のアプリケーションは、IBM zSystems プラットフォーム上のコア機能やデータにアクセスする際に、非常に大きな遅延が発生する可能性があり、コア機能やデータに頻繁にアクセスする場合にはスループットの低下にもつながります。

図 5 は、アプリケーションの配置場所によって、遅延が増加・低減できることを示しています。

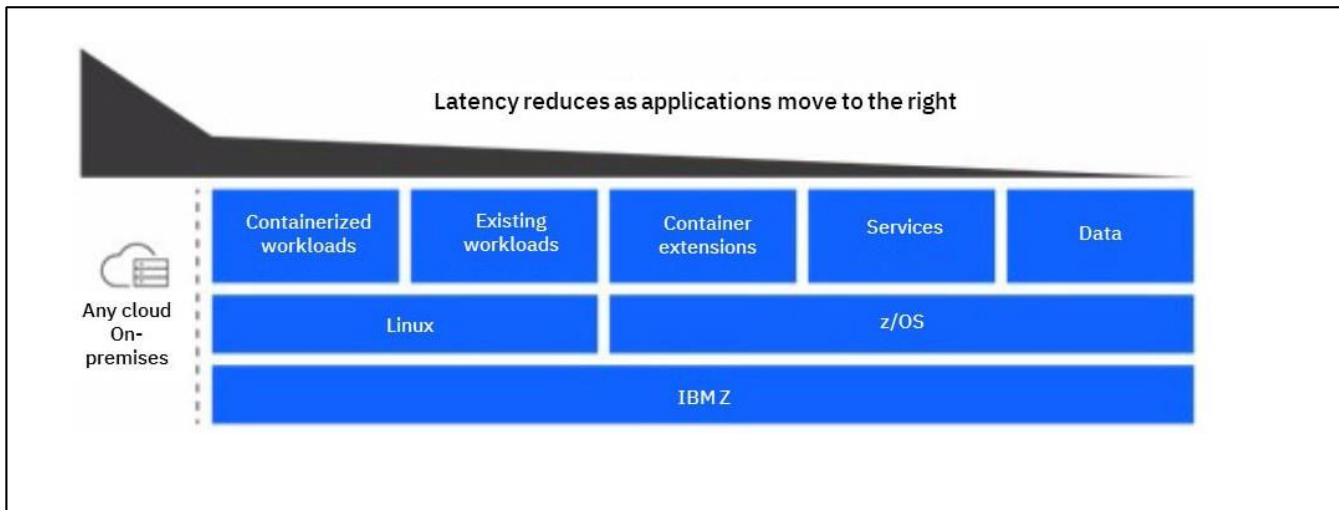


図 5 コロケーションによる遅延の低減

次の選択肢を検討してください。

- ▶ メインフレーム・データにアクセスする際に低遅延を必要としないアプリケーションには、あらゆるクラウド・プロバイダーのパブリックまたはプライベートクラウドが有効な選択肢となります。
- ▶ 低遅延を必要とし、z/OS に向かって「データ・グラビティ」があるアプリケーションは、多くの場合、[Linux on IBM zSystems](#) や [Linux on IBM zSystems 上の Red Hat OpenShift](#) 内にデプロイされます。IBM z/OS Container Extensions (zCX) は、z/OS ワークロードの一部として、Linux アプリケーションを z/OS 上に Docker コンテナとしてデプロイすることを可能にする z/OS 2.4 の新機能です。このアプローチを使用することで、z/OS 内の Linux 環境の操作制御を維持し、z/OS のサービス品質をアプリケーション・デプロイメントで活用することができます。個別の LPAR やシステム・イメージのプロビジョニングは必要ありません。Linux on IBM zSystems がすぐに利用できない場合に、アプリケーションを z/OS Container Extensions にデプロイすることもできます。これは、Linux on IBM zSystems アプリケーションと z/OS との統合に役立ちます。アーキテクチャーの例は、15 ページの図 6 を参照してください。

アプリケーション開発者は、z/OS アプリケーションやデータを使用して、人気の高いオープンソース・パッケージ、Linux アプリケーション、IBM ソフトウェア、サード・パーティー・ソフトウェアを開発し、データセンターで運用することができます。

- ▶ メインフレーム・アプリケーションとデータとのアフィニティーがある x86 上の既存のエンタープライズ・アプリケーションには、多くの場合 IBM zSystems プラットフォームが使用されます。分散アプリケーションを構成するより小さなコンポーネントが複数の小容量ハードウェア・プラットフォームにまたがって配置される分散システムとは対照的に、IBM zSystems プラットフォームと LinuxONE は、小さな物理フットプリントで極めて高いコンピュータおよび I/O 密度を提供し、垂直方向にも水平方向にもより優れたスケーリングを実現します。従来の分散ソフトウェアは、コア・ベースの料金体系が一般的で、コアの数が多いほどソフトウェアのライセンス・コストが高くなり、TCO 上昇の原因になります。よって、垂直・水平スケーリングによるアプリケーション・コンポーネントの統合やコロケーションは、パフォーマンスと TCO の両方を向上させます。
- ▶ データのグラビティとトランザクションのグラビティの両方により、グローバル・トランザクション・スコープで他の z/OS アプリケーションと接続する必要がある Java アプリケーションは、多くの場合、z/OS の IMS または CICS トランザクション・システムにデプロイされます。

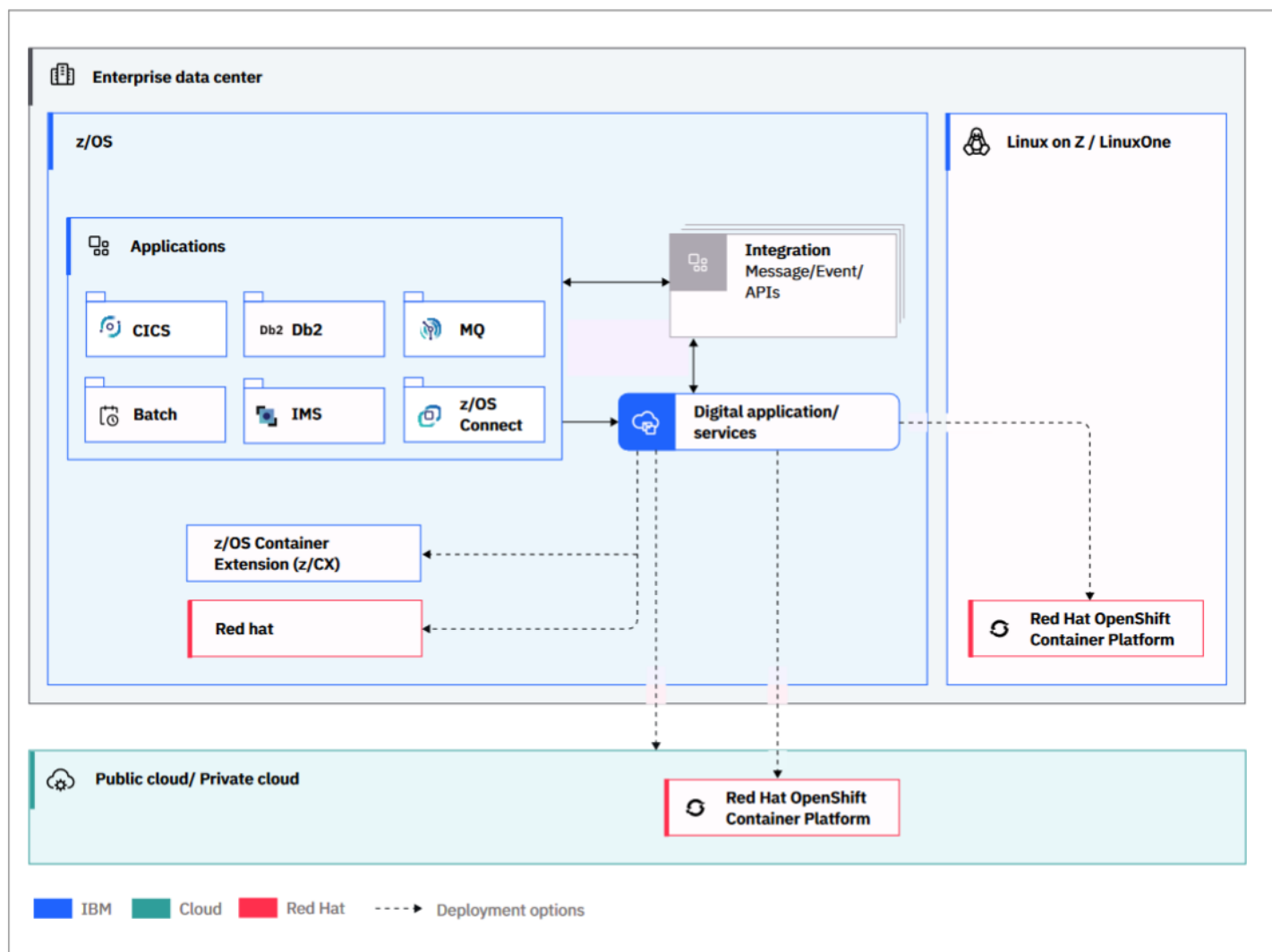


図6 z/OS、LinuxONE、およびクラウド上のアプリケーションが混在するエンタープライズ・データセンター

Ⅲ. ハイブリッドクラウドにわたる統合

ハイブリッドクラウド戦略を採用する企業は、アプリケーションの特性やサービス・レベル要件に応じて最適なプラットフォームを選択することができます。IBM zSystems プラットフォームとクラウドでビジネス・アプリケーションを使用・開発し、ハイブリッド・アプリケーションを作成するといったアプローチも可能です。

現在において、API は、標準的かつ制御された形で利用者にアプリケーション・サービスやデータを公開するための、最も安全で効率的な方法の一つです。充実した API エコシステムがあることで、企業はさまざまな利害関係者とコラボレーションすることができ、効率的なユーザー体験の提供とビジネス・プロセスの合理化が実現します。API 戦略の一部として多数のビジネス・クリティカル・アプリケーションを IBM zSystems プラットフォーム上で実行することは可能であり、またそうすべきです。そうすることで、企業は z/OS 上で稼働している安全なアプリケーションにアクセスし、機能の書き換えを行う必要なく、それらをクラウド上のデジタル・サービスに統合できます。

API、クラウド・サービス、および多くの業界にわたるトレンドにより、ビジネス・プロセスを変換してリアルタイムまたはほぼリアルタイムの応答を実現するというニーズが生み出されています。これは、情報を即座に把握する必要がない業界のユースケースにおいても同様です。イベント駆動型の迅速な意思決定と応答のシステムは、その用途が支払状況の確認、食事の宅配、出荷の追跡など何であって、顧客を満足させることができます。このような背景により、イベント駆動型アーキテクチャと情報処理のサブスクリプション・モデルが注目を集めるようになりました。

リアルタイム・データが大規模に必要とされる場合、コアのメインフレーム・システムは、増え続ける予測不可能な照会トラフィックを処理するために継続的に進化しなければなりません。多くの場合、データ照会用のモデルを、データ更新に使用されるモデルから分離することが最適です。さまざまなソースのデータと情報のキャッシング方法の詳細は、19 ページの『IV. 情報共有とデータ・アクセスの簡素化』で説明します。

API の使用

API は、アプリケーション間のアクセスを簡素化し、デジタル・エコシステムを拡大させる役割を担います。オープンな API は、少量のビジネス・データにリアルタイムかつオンデマンドで安全にアクセスして更新するためのソフトウェアにおける標準です。

API の実装とデプロイメントに関して、IBM zSystems プラットフォームは他のプラットフォームと何ら変わりありません。API は、プラットフォームに関係なく同じ機能と同じインターフェースを提供しますが、API の信頼性、拡張性、スループット、およびセキュリティは、API が利用されるプラットフォームに依存します。この点で IBM zSystems プラットフォームはその強みを発揮します。

IBM z/OS Connect は、IBM Z and Cloud Modernization Stack の一部として提供され、企業が次のことを達成できるようにします。

- ▶ 利用しやすい API を作成し、企業やクラウドのアプリケーションから z/OS アプリケーションとデータにアクセスできるようにする。
- ▶ z/OS アプリケーションから API を呼び出し、社内のエンタープライズ API で取得したデータ、またはクラウド・サービス (気象データ・サービスや市場データ・サービスなど) のパブリック API で取得したデータで z/OS アプリケーションを拡張する。

IBM z/OS Connect には、API 作成のためのローコード・アプローチを提供する Web ベースのデザイナーが含まれています。このデザイナーにより、既存のコードから API を作成したり、外部 API を使用したりできます。コンテナ化されたデプロイメントとツール・モデルは、企業の DevOps アプローチと完全に統合し、開発者が必要な API の構築に並行して取り組めるようにすると同時に、システム管理者がコンテナ・オーケストレーション・プラットフォームを使用して制御を維持できるようにします。

IBM z/OS Connect は、z/OS に直接デプロイすることも、コンテナ・マネージド・バージョンとしてデプロイすることもできます (17 ページの図 7 参照)。開発者は、企業の API ガバナンス・モデルをサポートし、ビジネス・アプリケーションで API をすぐに利用できるようにする「コントラクトファースト・アプローチ」を使用することにより、既存の OpenAPI 3.0 定義に基づく安全な API を作成できます。

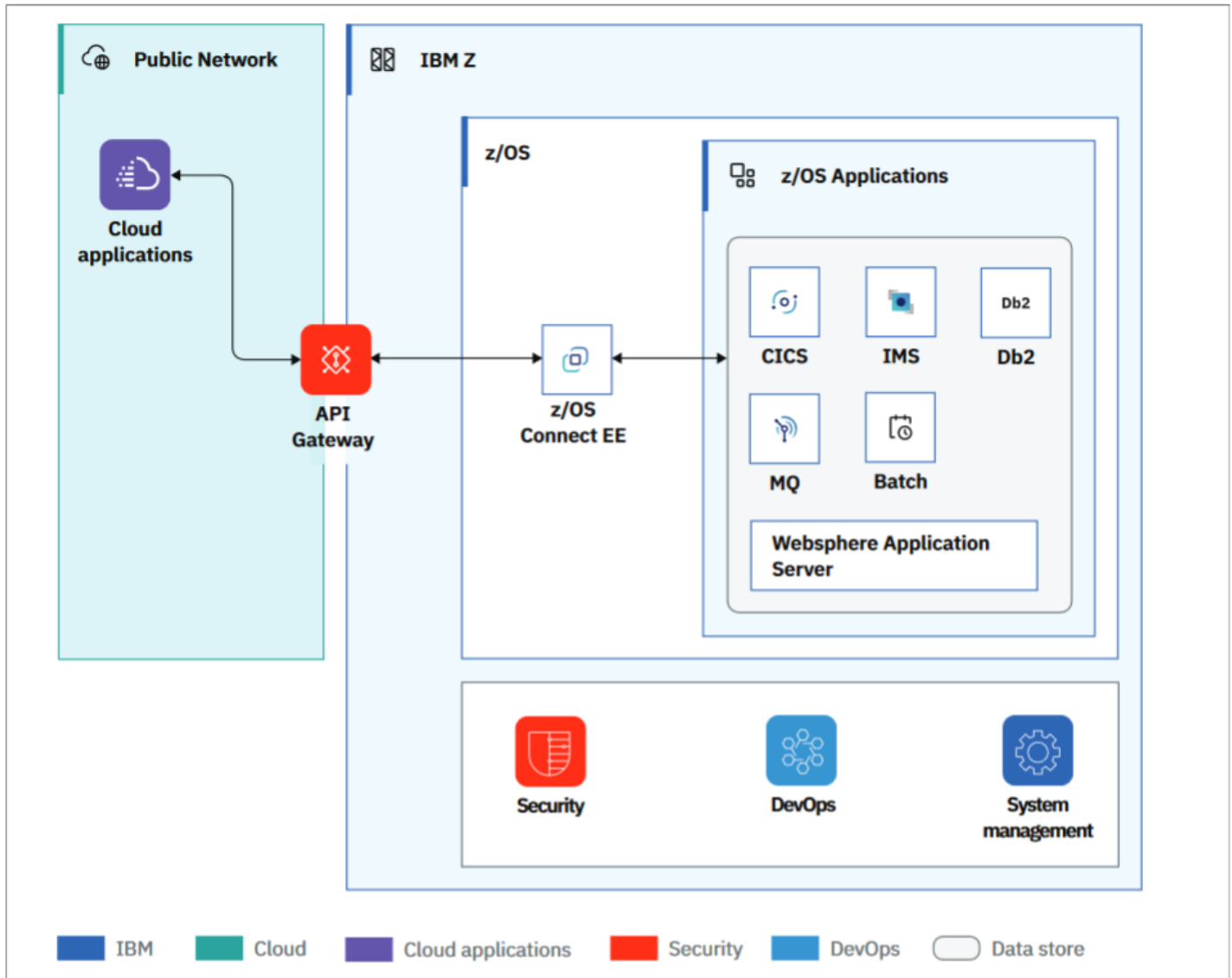


図7 クラウド上の API ゲートウェイ・アプリケーションと IBM zSystems プラットフォーム上のアプリケーションの使用

API のオーケストレーションおよび統制には、企業のハイブリッドクラウドで利用可能なものであれば、どの OpenAPI 3.0 標準 API ゲートウェイでも使用することができます。つまり IBM zSystems プラットフォーム上の API を、企業の API ポータルを通じて、企業内の他のすべての API と同じように管理、保護、および利用可能にすることができます。

イベント駆動型アーキテクチャの一部としてのメインフレーム

どの企業にも、意思決定や利用者への最新情報の提供と応答性の向上のためにリアルタイム・データを必要とする、クリティカルなビジネス・プロセスがあります。プロセスを構成するさまざまなイベントを効率的に伝達、実行、処理し、洞察やインテリジェンスを引き出せなくてはなりません。さらに、この情報を迅速で効率的、かつ柔軟な方法で共有し、意思決定や利用者への正しい状況の報告を促進するビジネス・プロセスに役立てる必要があります。

ビジネス・プロセスに適切な情報をリアルタイムで提供するために、多くの企業が企業全体にわたるイベント・ベースのアーキテクチャやバックボーンを実装しています。メインフレーム・アプリケーションはこのイベント・ベースのアーキテクチャの本質的な部分を構成します。その理由は、クリティカルなビジネス・データはアプリケーションと SoR (定型業務処理システム) から取得されるためです。

多くの場合、イベント駆動型アプリケーションには Apache Kafka が使用されます。IBM Event Streams や Confluent Platform などの Kafka ベースのイベント・バックボーンを、クラウドまたは Linux on IBM zSystems 上のメインフレームで実行することができます。

次に挙げるのは、Kafka と IBM zSystems ソフトウェアを使用することのメリットの一部です。

- ▶ z/OS Java アプリケーションは、Java Kafka クライアント・ライブラリーを使用することにより、イベント・データを直接 Kafka に共有できます。詳しくは、「Apache Kafka for IBM Z の設定 (Setting up Apache Kafka for IBM Z)」を参照してください。
- ▶ IBM MQ を流れる既存のビジネス・イベントやトランザクションのアーキテクチャーを必要とするイベントを、既存の資産への影響を最小限に抑えて再利用できます。イベントが IBM MQ に取り込まれた後、Kafka Connect の IBM MQ Source Connector を使用して Kafka にイベントが生成されます。
- ▶ 照会される情報の量が多く、さまざまなソースから集約される場合、IBM Z Digital Integration Hub (IBM zDIH) を使用して、既存データを最適化できます。

IBM zSystems プラットフォーム上のビジネス・アプリケーションも、次に挙げる類似のメカニズムを使用することにより、イベントのコンシューマーとして機能させることができます。

- ▶ Kafka クライアント・ライブラリーを利用することにより、z/OS 上の Java アプリケーションでイベントを使用できます。また、既存のアプリケーションでそれらのイベントを処理できます。
- ▶ Kafka Connect の IBM MQ Sink Connector を通じて、イベントを IBM MQ キューにストリーミングできます。ストリーミングされたイベントは従来の方法で使用できます。

図 8 はイベント駆動型アーキテクチャーの例を示しています。

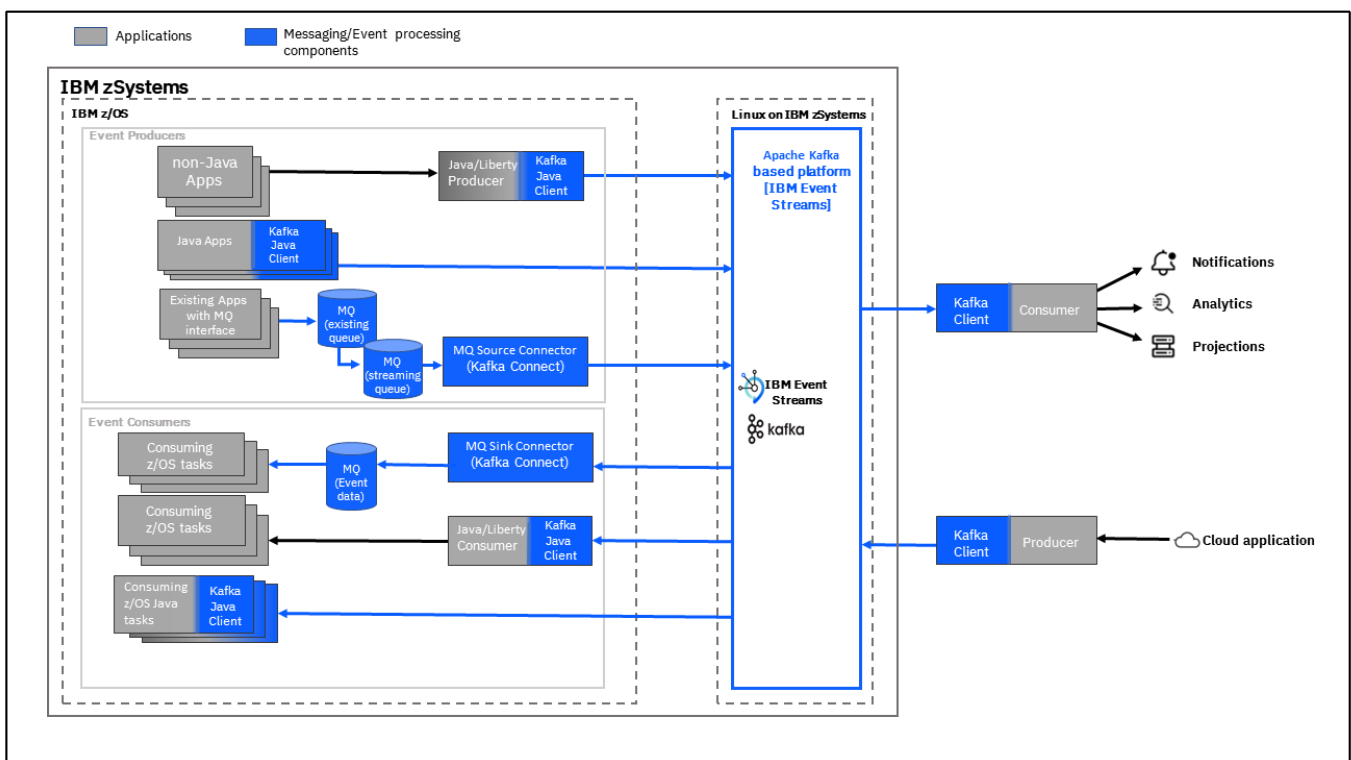


図 8 イベント駆動型アーキテクチャー

さらに、イベント情報が大規模に必要な場合には、IBM zDIH を使用する最適化された Command Query Responsibility Separation (CQRS) パターンを使用して Kafka のキューにフィードできます。

通常、Kafka クラスターは IBM zSystems プラットフォーム外で実行されますが、IBM Event Streams は、(クラウド・プラットフォームと分散型プラットフォームに加えて)、Linux on IBM zSystems 上の Red Hat OpenShift でクラスターをホストできます。クラスターを Linux on IBM zSystems 上で実行すると、IBM zSystems プラットフォームの高度なセキュリティおよびガバナンスの機能が利用できるようになることに加え、クラスターがメインフレーム・ワークロードに近づくことで、イベント・フローに対するレジリエンスとスケールが実現します。

イベントが Kafka トピックに取り込まれたら、それらを複数のクラウド環境にわたりストリーミングし、そのデータを通知や分析で使用したり、データベースで(射影、ストリーミング、または使用して)情報の「同期」コピーをローカルに作成したりすることができます。Kafka を通じたイベント処理の例には、トレンドのモニターや分析、デリバリー通知の送信や通知の処理、または注文、請求処理、口座の借方・貸方などのビジネス・プロセスに使用できるリアルタイム情報の取得などがあります。

IV. 情報共有とデータ・アクセスの簡素化

データはすべてのビジネスの基盤です。このエントリー・ポイントは、データを企業全体で収集、集計、サニタイズして、このデータをビジネス上の意思決定、機械学習、および分析に使用できるようにするというものです。

データは企業が所有する最も価値のある資産の一つです。ビジネスのニーズが拡大するのに従い、組織の誰もがデータへのアクセスを必要とするようになります。次に挙げるのは、データに対するニーズを示す数ある例の中の一部です。

- ▶ 営業やマーケティング部門は、市場および顧客行動の変化に迅速に対応しなければならない。
- ▶ 複数のチームが信頼性の高い財務予測を必要としている。
- ▶ 運用部門は洞察を使用して自動化を継続的に改善しなくてはならない。
- ▶ IT システム部門はアプリケーションのイノベーションとモダナイゼーションを実現しなければならない。

企業に蓄積されているデータの量は膨大で、一元化されたガバナンスと管理のメカニズムがなければ、いずれ管理しきれなくなります。データ・ファブリックを導入することで、データを編成し、データを動かすことなく適切なガバナンスとコンプライアンスに基づいてアクセスし、適切な目的のために効率よくふるいにかけることができます。

世界の構造化データのほとんどが、IBM zSystems プラットフォーム上で発生しています。そのため、IBM zSystems プラットフォームを含めたデータ・ファブリック・ソリューションを活用すれば、最新のトランザクション・データに基づくより優れたビジネス上の意思決定が実現します。また、メインフレームから得た構造化データと非構造化データを意思決定に活用できるようにすることで、顧客中心主義を強化し、データの適切なセキュリティとガバナンスも確保できるようになります。

IBM zSystems プラットフォーム上に存在するデータ・モダナイゼーションの機能により、次に挙げる目標が達成できるようになります。

- ▶ メインフレームの構造化と非構造化の両方のデータへのオープンなアクセスと、大規模なビジネス・インテリジェンスおよびレポート作成のためのファブリックへのデータの複製を通じて、IBM zSystems プラットフォームをハイブリッドクラウド・データ・ファブリックに組み込む。
- ▶ ビジネス・ソフトウェアの情報共有を、大規模なデータ共有に対応できるようにする。その際に、共有されるデータに基づいて動作する他のビジネス・プロセスに影響が及ばないようにする。

データ・ファブリックの一部としての IBM zSystems プラットフォーム

データ・ファブリックとは、適切なビジネス・コンテキストで、適切なガバナンスに従って、適切なデータを適切な人にインテリジェントかつ自動的につなげるアーキテクチャーです。これは、データを説明する(注釈を付ける)メタデータによって達成されます。データ・ファブリックは、既存のデータ管理インフラストラクチャーの総入れ替えを必要としません。既存のシステムは、メタデータを共有することにより、ファブリックに受動的に参加できます(図9参照)。

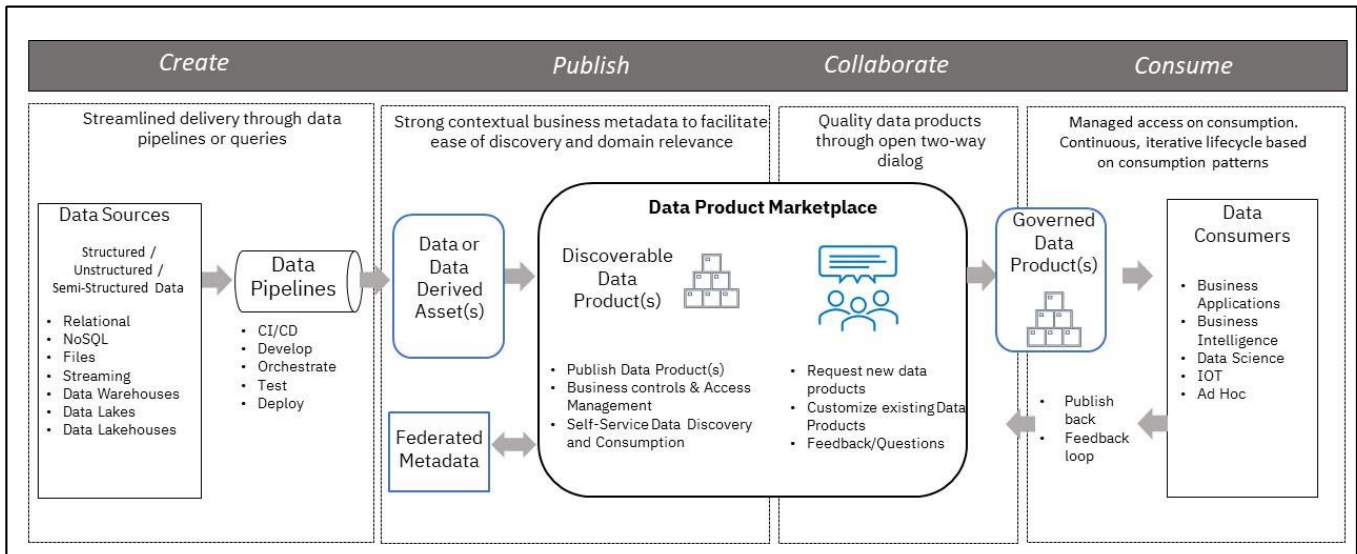


図9 データ・ファブリック・アーキテクチャーの作成、パブリッシュ、コラボレーション、使用

データ・ファブリックの主要な目的は、機械学習、ビジネス・インテリジェンス、分析であろうと、ビジネスが適切な意思決定のために、適切なデータを使用できるようにすることです。

優れた設計のデータ・ファブリックは、次に挙げる目標を達成できます。

- ▶ ハイブリッドクラウド全体に分散したデータへのアクセスの簡素化。
- ▶ データ・ガバナンスとプライバシーの自動化。
- ▶ 360 度ビューを提供するためのドメイン間のデータ統合。
- ▶ リアルタイムで異常データを除去するためのデータのクレンジング。
- ▶ AI のトレーニング・モデルの提供。

IBM zSystems プラットフォームのデータをデータ・ファブリックに統合することで、IBM zSystems データの全体像と拡張知識がビジネス・ユーザーに提供されるようになります。また、データウェアハウスやビジネス分析のために IBM zSystems プラットフォームから複製されたデータにアクセスする際に必要なガバナンスとセキュリティも実現します。

これらのプロセスの多くで収集・編成されるデータはメインフレームから取得されます。そのため、包括的なデータ管理システムには、さまざまな異種のデータ・ソースからデータを取り込むための堅固なメカニズムが必要です。

データ収集の簡素化

IBM Cloud Pak® for Data は、データの収集、編成、分析を簡素化・自動化し、企業全体で AI 活用を加速します。

IBM Cloud Pak for Data で異種のさまざまなデータ・ソースのデータにアクセスするための基盤となるのが IBM Watson® Knowledge Catalog です。

Watson Knowledge Catalog は、検出とガバナンスの機能を提供し、IBM Data Virtualization Manager for z/OS および IBM Db2 for z/OS に格納されているメタデータを使用して、z/OS のあらゆる異種データ・ソースをカタログ化します。IBM Data Virtualization Manager for z/OS は、z/OS 内のすべてのデータの仮想の統合ビューを提供します。これにより、データを移動、複製、変換する必要なく、アプリケーションが IBM zSystems にあるデータに、そのデータがある場所で読み取り/書き込みアクセスを行えるようになります。

Db2、Virtual Storage Access Method (VSAM)、Information Management Systems (IMS)、Adaptable Database System (ADABAS)、Integrated Database Management Systems (IDMS) などのリレーショナルおよび非リレーショナルのデータは、HTTP、SQL、および REST (z/OS Connect 経由) を通じてアクセスできます。IBM Data Virtualization Manager for z/OS は、これらの異種ソースをデータ・ファブリックの一部として使用できるようにします。図 10 は、仮想化の使用によりデータ収集が簡素化される仕組みを示しています。

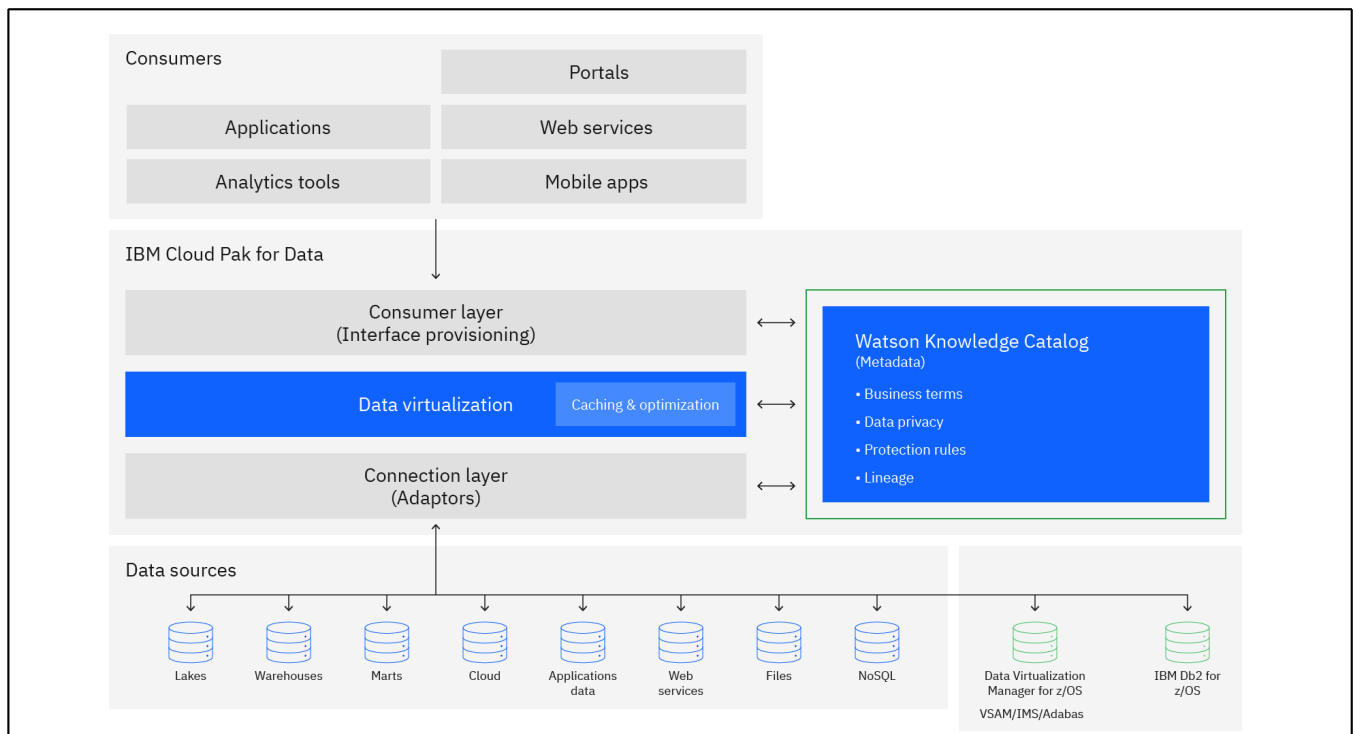


図 10 仮想化を通じたデータ収集の簡素化

Watson Knowledge Catalog 内の、IBM zSystems データを含めた一元化されたデータのビューを使用して、任意のプラットフォームで機械学習 (ML) モデルを構築、テスト、トレーニングできます。その後、AI モデルを Watson Machine Learning for z/OS にデプロイし、z/OS 上で稼働するビジネス・サービスの複雑な情報のニーズに対応できます (図 11 参照)。

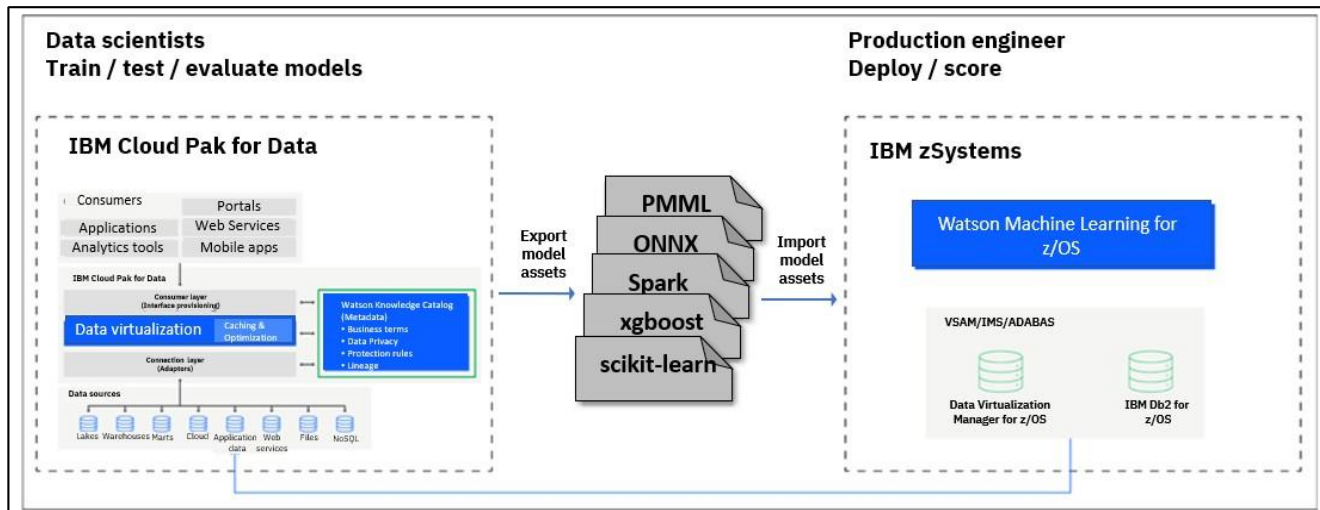


図 11 ビジネス・サービスの複雑な情報に対応するための ML モデルの使用

21 ページの図 10 および図 11 に示すとおり、データ・ソースには、レイク、ウェアハウス、マート、クラウド、アプリケーション・データ、Web サービス、ファイル、NoSQL が含まれます。

リアルタイムの情報の大規模な最適化

デジタル・トランスフォーメーションでは、API とイベントの両方を活用して、コアのビジネス処理との有意義な対話を作成し、エコシステムを拡張します。イベント処理は、最新の情報に基づく効率的な意思決定を可能にし、全体的なビジネスの成果を向上させることができます。

リアルタイムの情報が大規模に必要な場合には、データ・ソースに直接アクセスすることは推奨されません。コア・アプリケーションが影響を受ける、最新のデータや整合性のあるデータを手でできない、ハイブリッドクラウド・アプリケーションで簡単に利用できないデータ・モデルがある、完全な情報やコンパイルされた情報が確認できない(口座残高など)といった可能性があるためです。

これらは、増え続ける予測不可能な照会をクリティカルなアプリケーションから分離することが推奨される理由を示す例です。企業は、日常業務がそうした照会から影響を受けないようにするために、大規模な予測不可能な照会を分離する、最適化された Command Query Responsibility Segregation (CQRS: コマンド・クエリー責務分離) パターンを実装する必要があります。

目標は、デジタル・トランスフォーメーション、イベント処理、クラウド・サービスのために、大規模にリアルタイムのデータにアクセスできるようにすることです。

IBM Z Data Integration Hub

IBM Z Digital Integration Hub (zDIH) は、IBM zSystems プラットフォーム (z/OS) 上でホストされているコアのビジネス・システムと、それらを使用する、ハイブリッドクラウド上でホストされているアプリケーションとの間でリアルタイムの情報フローを提供します。IBM zDIH は、アプリケーションが大規模なリアルタイムのメインフレーム・データを必要とする場合に、最適化されたパフォーマンスを提供し、コア・アプリケーションへの影響を最小限に抑えます。

IBM zDIH には、Java ベースのラインタイム環境とインメモリー・データベースを使用する高性能キャッシュ、ローコードの導入を加速するためにキャッシュやアプリケーションを自動生成する IBM zDIH 開発者キット、テンプレート、およびサンプルが含まれており、アプリケーション出口、アプリケーションからの直接書き込み、または IBM MQ on z/OS の使用を通じて、CICS および IMS アプリケーションとの統合を加速します。

図 12 は、IBM zDIH を適用して最適化された CQRS パターンを実装し、ハイブリッドクラウドと統合されたメインフレームのコア SoR を加速する方法を示しています。

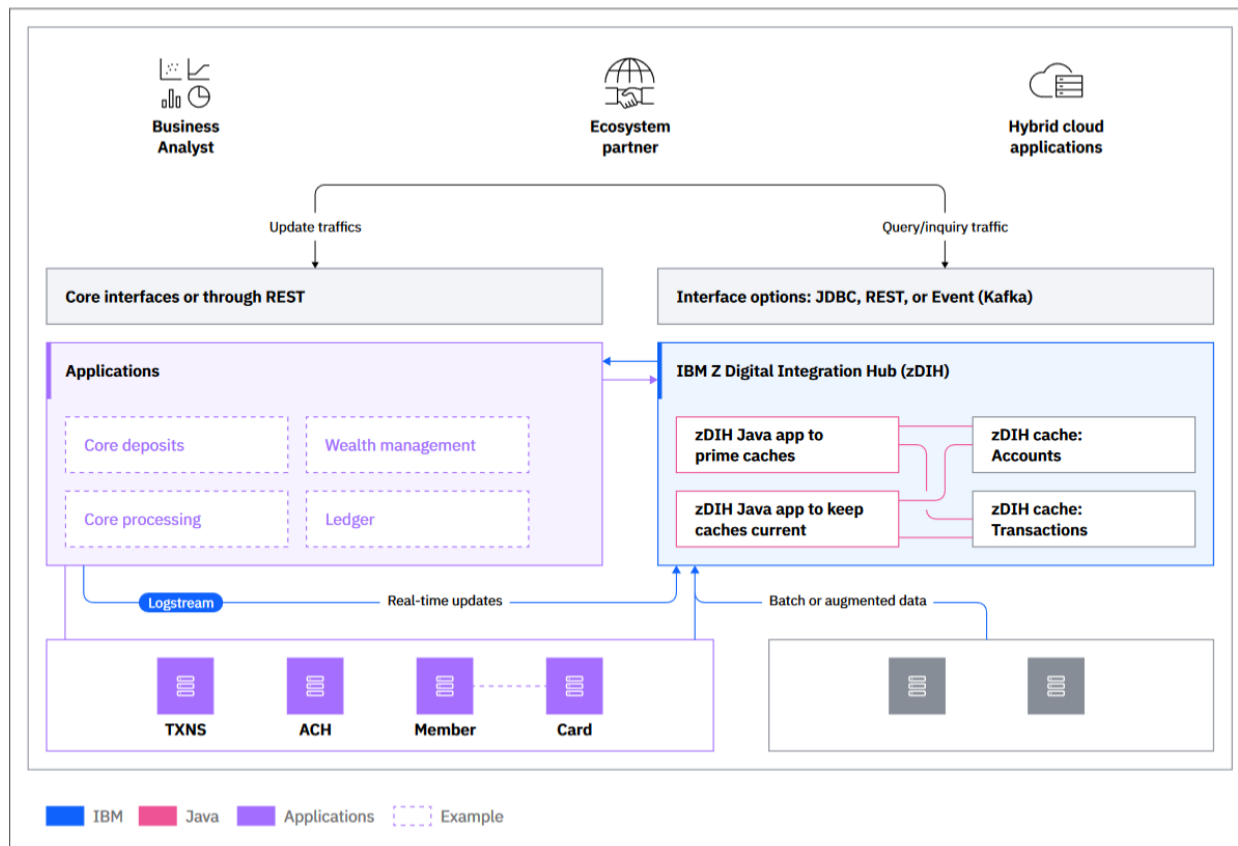


図 12 IBM zDIH の使用による最適化された CQRS パターンの実装

zIIP プロセッサ上で IBM zDIH を実行することで、SoR ですべての照会トラフィックに対してトランザクションを開始する必要がなくなり、また、変更されていない可能性のある情報が再計算されなくなります。その結果として最適化が実現します。

IBM zDIH は、イベント処理 (Kafka 経由)、API 管理、またはパブリッククラウド上のメディエーションのために、キャッシュ情報を企業全体のアーキテクチャーに直接流すことができます。Kafka トピックは、キャッシュの情報が変更された後に更新されるように構成できます。Kafka トピックは、リアルタイムという特性から多大な価値を引き出すパブリッククラウド (SaaS) ソリューションで使用でき、クラウドネイティブ・アプリケーションの価値実現までの時間を短縮できる可能性があります。

IBM が提供するデータ・モダナイゼーション手法を使用することにより、メインフレームに含まれている貴重なデータを、企業のデータとイベント処理のエコシステムの一部とすることができます。

- ▶ リアルタイム情報のニーズ (ビジネス・イベント処理、クラウド・デジタル・サービス、エンタープライズ・アプリケーションとサービスなど) のために、メインフレームのデータと情報はプラットフォーム内にとどめ、IBM zSystems プラットフォームから直接アクセスできるようにすべきです。ベスト・プラクティスとして、デジタル・サービスやクラウド・サービスが少量の情報にほぼリアルタイムでアクセスし、それを更新する必要がある場合は、API を使用します。
- ▶ データ・ファブリックは、ハイブリッドクラウド全体にわたり分散したデータへのアクセスを簡素化し、データ・ガバナンスとプライバシーを自動化し、ドメイン間でデータを統合します。IBM Cloud Pak for Data は、データがある場所でデータにアクセスし、IBM zSystems プラットフォームのデータ・ソースをファブリックに統合できます。メタデータ・カタログ自体は、任意のクラウド・プロバイダーの IBM Cloud Pak for Data の一部として Red Hat OpenShift 内に含めることができます。
- ▶ IBM zSystems プラットフォームには、分析やビジネス・インテリジェンスに必要なデータウェアハウジング機能が装備されています。ベスト・プラクティスとして、データをエンタープライズ・データレイクまたはウェアハウスのある場所に配置します。データレイクまたはウェアハウスがメインフレーム上にない場合、IBM は、適切なセキュリティを使用し、読み取り専用でデータを複製できる、ビジネス分析のための安全なツールを提供できます。

まとめると、図 13 で示すとおり、IBM zSystems プラットフォームは企業のデータ・ファブリックの一部として機能し、最新のデータ・アクセスを実現するのに役立ちます。また、IBM Data Virtualization Manager for z/OS を利用することにより、アプリケーションからのすべてのデータへの形式を問わないオープンなアクセスを提供します。

IBM Db2 Analytics Accelerator を利用して照会を高速化することにより、SLA に影響を及ぼすことなくトランザクションと分析の同時実行が実現します。IBM zSystems プラットフォームは、イベント駆動型アーキテクチャーの一部として IBM zDIH を利用して CQRS パターンを実装することにより、大規模なリアルタイムの情報とイベント処理を必要とするチームにデータへのアクセスも提供できます。

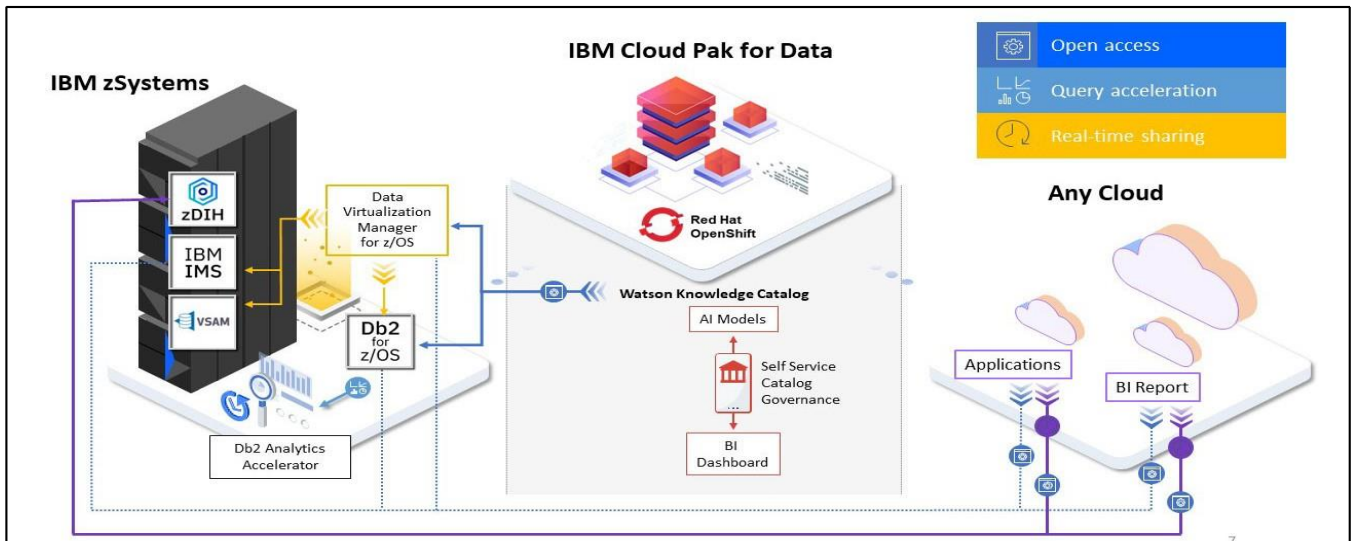


図 13 IBM zSystems と IBM Cloud Pak for Data およびクラウドとの統合

V. 企業全体の DevOps による俊敏性の向上

最新の DevOps プラクティスを導入し、最新の開発/運用体験を実現することで、メインフレーム上のアプリケーション開発と管理にまつわる不要な特殊性を取り除き、新しい人材をプラットフォームに引き込みやすくします。

アプリケーション・モダナイゼーションのエントリー・ポイントである IBM の DevOps へのアプローチとその機能は、トランスフォーメーションの原則一式と Git を土台とした最新のオープンソース・ベースのソリューションに基づいて構築されています。このアプローチは、企業が、プラットフォーム・ニュートラルな、企業全体で共通のガバナンス・モデルを使用して、エンタープライズ・アプリケーションを構築できるようにします。

メインフレーム開発における DevOps プラクティスは、品質を担保しながら、テスト環境の構成とデプロイメントを自動化することにより、リリースサイクルを年に数回からオンデマンド（1日に複数回）に短縮することで、ビジネスの俊敏性を高めることができる可能性があります。

IBM の DevOps ツールは、メインフレーム・アプリケーションを企業全体の CI/CD パイプラインに組み込みます。デプロイ後は、IBM zSystems プラットフォーム上で稼働するコンポーネントを含めて、IBM はハイブリッドクラウドに対する一貫したアプリケーション可観測性を提供できます。

文化としての DevOps

DevOps とは、継続的な学習や改善、イノベーションの促進、透明性や可視性の確保、および信頼といった文化を醸成するための取り組みです。大規模な企業が本気で改革を推し進めたいのであれば、新しい習慣や文化を推奨し発展させる必要があります。そうすることで、イノベーションを加速し、市場の需要に対応できます。このアプローチには、ソフトウェア開発チームが最新のツールを採用し、自動化を実装することも不可欠です。IBM zSystems プラットフォームは、企業の DevOps 戦略の一部として活用することができます。

DevOps 文化は、企業がレジリエントなソフトウェア・デリバリー・プロセスを構築し、あらゆるスキル・ギャップに対応できるよう後押しします。単一の異種パイプラインを使用して、IBM zSystems プラットフォーム、クラウド、または企業の他の部分などの、適切なセキュリティやガバナンスが確保され、品質ゲートが設定された複数のターゲット・プラットフォームと環境にまたがり、アプリケーションの開発、統合、およびデプロイをオーケストレーションすることが必要です。

開発プロセスと同様に、IT 運用においても IBM zSystems プラットフォームをエンタープライズ・アプリケーションの可観測性の一部として含めることが推奨されます。IT 運用への AI と可観測性の統合は、IT 運用の最新化を目指し、激化するビジネス競合状況で先手を打ちたいと考えている企業にとって重要な検討事項になっています。

可観測性は、企業がハイブリッドクラウド・アプリケーションのパフォーマンス、信頼性、および拡張性を効果的に管理するために必要な可視性と理解を提供します。AI を活用するソリューションをデプロイすることで、自社 IT インフラストラクチャの増大し続ける複雑性と規模を効果的に管理し、チームのスキル・ギャップに対応し、コストを削減しながら SLA を向上できるようになります。

定型タスクを自動化し、洞察を統合することにより、IT チームは価値の高い作業に専念でき、時間とリソースが解放されることでイノベーションを促進し、サービス・デリバリーの全体的な品質を向上できるようになります。

さらに、AI を活用してダウンタイムや停止に対処することにより、リアルタイムで問題を特定し、解決できます。これは、ユーザー体験と顧客満足度が向上につながります。

IBM zSystems プラットフォームの DevOps と CI/CD

IBM では、DevOps プロセスに関し、ツールとプロセスを企業全体で標準化し、必要な部分にのみ IBM zSystems プラットフォーム固有の機能を使用するというビジョンと戦略を掲げています。このアプローチを次に説明します。

- ▶ プラットフォーム固有のツールが企業の統一された開発プロセスを妨げる領域では、オープンソース・ツールや企業の標準ツールを採用したり、z/OS を使用するよう拡張したりする。この取り組みには、ソースコード・マネージャー (SCM)、エディター、パイプライン・オーケストレーター、統合テスト自動化、成果物リポジトリの企業の標準ツールを採用することが含まれます。
- ▶ 単体テストの自動化、構築、および従来の z/OS アプリケーションのデプロイメントなど、プラットフォーム固有でなければならない領域で IBM zSystems 機能を開発する。

このアプローチを使用することで、企業のオンプレミスとクラウドにまたがり同じ DevOps ツールをメインフレーム開発で使用できます。

IBM zSystems プラットフォームは、多くのプログラミング言語、ツールチェーン、および開発プラクティスをサポートします。最新の開発には統合開発環境 (IDE) が使用されます。IBM の z/OS 開発における戦略は、「IDE の持ち込み (bring-your-own-IDE)」の概念を中心に据えています。Eclipse、Visual Studio、[Red Hat OpenShift Dev Spaces](#) 上のコンテナ・ベース Web IDE などの IDE を持ち込むことで、開発者は使い慣れた言語とツールを使用してメインフレームの分析とイノベーションが行えるため、ビジネス・ロジックの実装に集中できます。

メインフレーム・アプリケーションは数十年かけて開発されたものがほとんどであるため、最近開発されたアプリケーションと同じレベルの知識が保たれていない場合があります。分析を実施することで、開発者、アーキテクト、およびビジネス・アナリストが、コードの所有権を取り戻し、異なる部分がどのように相互作用するかを把握し、関与するデータ・モデルを理解するのに役立ちます。

このような分析により、文書化されている情報とは異なる可能性が高いアプリケーションのバージョンに関する最新の情報も提供されます。IBM ADDI は、アーキテクトと開発者が、アプリケーション・フローを視覚化し、レポートを生成して自社のモダナイゼーション戦略に基づいて行動し、自信を持って増分を計画できるよう、メインフレーム・ソフトウェアのアセットとリソースに関する最新の利用しやすい情報を提供することにより、スキルと知識のギャップを埋める支援をします。

IBM が委託した調査「IBM のアプリケーション・ディスカバリー・ツールで始めるメインフレームの DevOps ジャーニー - Sun Life 社はモダナイゼーションを通じてデジタル・トランスフォーメーションを加速 (A Mainframe DevOps Journey Starts with IBM Application Discovery Tools-Sun Life accelerates digital transformation through modernization)」では、次のように述べられています。

「デジタルな競合他社に対抗し、お客様の期待に応えるためには、市場投入までの時間が極めて重要です。当社の成果から、ADDI (IBM Application Discovery and Delivery Intelligence) のような DevOps ツールを使用することで、長期的に見てメインフレームで分散環境と同等の速度が実現できることが実証されました」

記事はさらにこう続けています。

「……開発者は ADDI を使用して、さまざまな投資アプリケーションと保険アプリケーションとの間の複雑なつながりを見つけ出しました。1 行ごとのコード検索に比べ、かかった時間は半分でした」

「開発者は、コードのつながりを特定することに加え、変更が COBOL のルール・ベースの制限に従っているかどうかを知る必要がありました。ADDI を使用することで、分析時間を 35% 削減することができました」

継続的インテグレーションとは、複数のチームが分離された環境で段階的に開発を行い、コードを継続的に統合し、増分をテストできるようにするプラクティスです。そこで、最新の SCM が、どのような継続的インテグレーション・パイプラインにも不可欠になっています。

Git のような最新の SCM を使用することで、エンタープライズ・アプリケーションを構成するすべてのテクノロジーの設定、自動化、および管理を単一の SCM で行うことができます。それにより、企業全体にわたるパイプライン・プラクティスの監査性と標準化を実現できます。最新の SCM は、従来の z/OS ベースの SCM とは異なり、並行開発と制御されたコードのマージをサポートします。このアプローチにより、開発者は変更のコーディングや単体テストを分離された環境で実行することができます。z/OS アプリケーションの記述には、COBOL、PL/I、アセンブリ言語、JCL、Java、Go、Python などの言語がよく使われます。データベースとミドルウェアの構成は、Git ベースのパイプラインでファースト・クラスとして扱うことができ、成果物リポジトリにデプロイ可能なバイナリー・ファイルが保存されます。

SCM を使用することにより、アプリケーションは一度構築すれば何回でもデプロイできます。これにより、ソースコードの追跡可能性が実現し、複数の環境にまたがってバージョンをデプロイできるようになります。Java および分散型テクノロジーにおいて、Maven のようなビルド・システムは、ビルドに必要なすべての依存関係を紐づけるのに役立ちます。IBM Dependency Based Build は、必要な依存関係を理解し、Git で従来のメインフレーム・アプリケーションを構築するための最新のビルド・ツールキットを提供します。

継続的インテグレーションでは、ビルドされた成果物に回帰がないことを担保するために、パイプラインの一部として新規コードが自動的にテストされる必要があります。開発のさまざまな段階でテスト自動化を組み込むことは、CI/CD の鍵です。

ビルド・フェーズの一部としてデプロイメント前のテストを自動化することで、開発者が外部の影響や変化ではなく、コードの更新に集中できるようになります。多くの場合、Java や Python などの最新の言語には、コア言語の一部として、またはアクセス可能なフレームワークとして利用可能な単体テストのフレームワークが備わっています。従来の z/OS アプリケーションに対しても同じ方法で単体テストを実行できます。IBM z/OS Automated Unit Testing Framework (zUnit) と IBM Z Virtual Test Platform を使用することで、パイプラインの一部として、Customer Information Control System (CICS) や Information Management System Transaction Manager (IMS TM)、データベース呼び出しなどのランタイムをスタブし、環境に依存しない方法で自動テストを作成・実行することができます。z/OS 開発者は、COBOL や PL/I アプリケーションの変更を共有環境にデプロイする前に、それらの変更に対し、ブランチで単体テストを分離して行えます。これにより、シフトレフトによるアプリケーションのテストが実現します。ランタイム環境がスタブされるため、これらのテストはデータをリセットする必要なく、繰り返し実行することができます。単体テストは SCM で管理可能で、企業の CI/CD パイプラインの一部として自動で実行できます。

アプリケーションの変更がビジネス機能に統合されるに従って、自動化の要件は、言語固有であることから離れて、アプリケーションの外部インターフェースを合わせて統合できることに移行します。

そのため、テストはプラットフォームをまたいで拡張できなければなりません。デプロイメント後のさまざまなレベルのテストに対応し、SoE (協働のための情報活用システム) を促進しながら、変更を SoR で検証する必要があります。オープンソース自動化フレームワークを提供する IBM の Galasa は、メインフレーム・アプリケーションとのシームレスな統合を実現します。ほとんどのお客様がすでに、一連のデプロイメント後テストを手動で実行しています。これらのテストを自動化することにより、アプリケーションのデプロイメントを加速できます。また、テスト担当チームがさらなるテストを実施して、一層品質を向上できるようになります。

例えば、REST API を通じて COBOL アプリケーションからコア・データを取り出す Node.js 上のハイブリッド・アプリケーションは、z/OS Automated Unit Testing Framework (zUnit)、IBM Z Virtual Test Platform、および Galasa を使用して、エンド・ツー・エンドでテストできます。zUnit と IBM Z Virtual Test Platform を使用することにより、COBOL プログラムの機能拡張の単体テストを簡単に、分離して実行できます。変更がマージされ実動環境へのリリース準備が実行された後、トランザクション、API、および Node.js のデプロイ後の統合テストを、Galasa を使用して自動化できます。

アジャイルな開発環境

継続的テストの一部としてオンデマンド環境が提供されます。オンデマンド環境があることにより、開発チームは、テスト・システムを中断したり、環境が割り当てられるのを待ったりする必要なく、分離された環境で変更をテストできます。IBM Cloud またはオンプレミスの Red Hat OpenShift で必要に応じて z/OS 環境のスピンアップやスピンドアウンができるようになったことで、真にアジャイルな z/OS 開発が実現するようになりました。

目的に応じて独立したテスト環境を作成するという考え方は、メインフレーム開発のパラダイム・シフトです。このアプローチは、共有環境という問題を解決し、z/OS の開発者体験を他のコンピューター・プラットフォームに近づけます。

これらの環境は、IBM Wazi as a Service を使用する、IBM Cloud 上の IBM Z Cloud and Modernization Stack、および Linux on IBM zSystems 上で動作する IBM Virtual Dev and Test for z/OS の一部として利用できます。これらの製品は、ビジネスの変更がマージされてさらなるテストが行われる前に、自動化されたパイプラインを使用して開発者ブランチとフィーチャー・ブランチの独立した単体テストが正常に行われることを確実にする方法を開発チームに提供します。

図 14 は、z/OS アプリケーションが企業全体の DevOps パイプラインにおいてアクティブな役割を担うことができることを示すアーキテクチャーの例です。

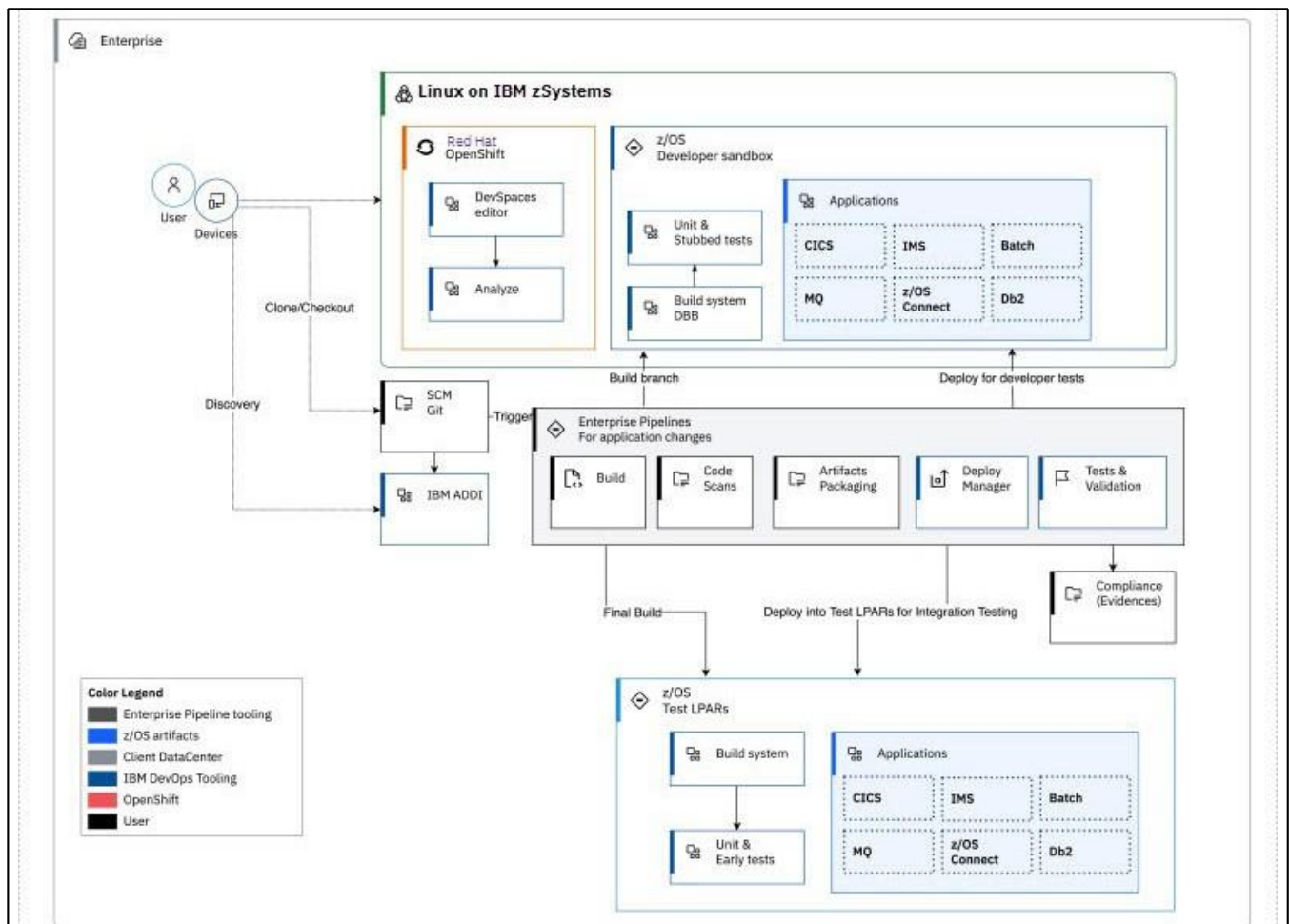


図 14 企業の DevOps パイプラインにおける z/OS アプリケーション

DevOps ジャーニー

メインフレーム・アプリケーションの開発チームは、企業内の異なるプラットフォーム間に存在するサイロを解消し、ベスト・プラクティスを共有・導入することにより、DevOps トランスフォーメーションを開始できます。最新の開発体験を始めるには、反復的に、つまり一歩ずつ着実に進めていくことが大切です。バリュー・ストリーム分析のアプローチは、開始するにあたっての最適な場所を判別するのに役立ちます。何が評価対象の企業にとって最大の価値をもたらすかによって、開始点は新しい IDE であったり、テストの自動化であったりするでしょう。ほとんどの企業向けの反復的な DevOps ジャーニーの例には、次の手順が含まれます。

1. Visual Studio Code (VSCode)、Red Hat Dev Spaces、Eclipse などの最新の開発ツールを採用し、新規開発者がコードの編集、理解、デバッグを行えるようにする。テスト環境がないことが懸念であれば、サンドボックス環境の概念を導入する。
2. 並行して、統合テストとシステム・テストの自動化の機会を特定する。
3. 分散型システムですでに導入されている現行の企業ベスト・プラクティスと既存ツールに基づく開発パイプラインを設計する。
4. ブランチを独立させることができる Git ベースの最新の開発プロセスにメインフレーム・アプリケーションを反復的にオンボードする。ブランチから構築できるベーシックなパイプラインを作成し、既存の環境にデプロイする。
5. 開発チームに単体テストの概念を導入する。パイプラインのビルド・ステップに単体テストを追加する。
6. コード・スキャン、自動テスト、自動監査レポートなどのゲーティング・メカニズムを含めるようパイプラインを拡張する。

業界標準の自動化テクノロジーと言語 (Red Hat Ansible や Python など) は、パイプライン内の自動化やインフラストラクチャのプロビジョニングのために、IBM zSystems プラットフォームとその他のプラットフォームにまたがり一貫して使用できます。これらのツールは、企業内で使用されている既存のメインフレーム自動化ツールにリンクすることができます。これらの機能すべてが、z/OS 開発体験と他のクラウドネイティブ開発体験との連携を実現し、z/OS アプリケーションの継続的インテグレーションを確実なものにします。

最後に、監査・制御されたデプロイメントは、CI/CD プロセスの重要な一部です。Red Hat Ansible に基づいてスクリプト化されたデプロイメント・フレームワークは、z/OS アプリケーションにも同じように適用できます。デプロイメントをクラウドや Kubernetes ベースの環境から管理して、IBM zSystems プラットフォームを最新のデプロイメント・エコシステムに組み込むことができます。

アプリケーション可観測性

DevOps ジャーニーに投資することに加えて、ハイブリッド・アプリケーションを管理するための適切なツールとプロセスを配備することも大切です。IBM zSystems プラットフォームを含めたハイブリッド・アプリケーション可観測性を実現するための最初のステップとして、次の 3 つの基本的な原則があります。

- ▶ ハイブリッドクラウド統合を通じて IBM zSystems プラットフォームの**可視性を向上**し、早期の問題解決、迅速な意思決定、効率向上に役立てる。⁹ これは、モバイルからメインフレーム・アプリケーション・トランザクションに至るまでのトレーシングの有効化、主要なインフラストラクチャのメトリックやログに対するゴールドデン・シグナルのストリーミング、ハイブリッド・アプリケーション環境全体での監視対象イベントと異常の相関により達成できます。
- ▶ プロアクティブなインシデント検知、自動相関、および高度なログとデータの分析を通じた AI の洞察で**運用チームを強化**する。運用チームは、機械学習を活用して履歴データから通常の運用を理解することで、主要業績指標のしきい値の設定を自動化できます。また、異常の相関と最新の分析プラットフォームを使用して根本原因を切り分けることで、診断時間を短縮できます。
- ▶ ワークフロー自動化を通じて**インシデント管理を迅速化**し、定型タスクや反復タスクの処理時間を短縮して、AI 主導の洞察を現場の従業員に提供する。チャット・ベースのコラボレーションとインテリジェントな自動化を通じて、既知の IT 運用の問題に完全に自動化された修復を使用できるようにすると同時に、新たな問題を透過的にトリガーできるようにして、制御室に大きな影響が及ぶ状況を回避します。

⁹ <https://www.ibm.com/z/hybrid-cloud>

より迅速な問題解決

1 つ目の原則では、システム全体のトランザクション・フローを自動的にトレースできるようにする動的なトレーシング・テクノロジーを使用します。これにより、基礎の IBM zSystems インフラストラクチャーを含むさまざまなシステムにわたるアプリケーションの振る舞いとパフォーマンスの完全なビューを提供します。これを基に、SRE は問題点を特定し、ハイブリッド・アプリケーションの問題を切り分けることができます。

IBM OMEGAMON® 製品などの IBM zSystems インフラストラクチャー監視ソリューションとの統合を通じて、主要なインフラストラクチャーのメトリックが収集・分析され、IBM Instana® にストリーミングされます。これにより SRE は、複数のハイブリッド・アプリケーションに影響を及ぼす可能性のある広範囲なインフラストラクチャーの問題があるかどうかを判別できます。

プロアクティブなインシデント管理のためのハイブリッド・アプリケーション・アプローチをとる場合には、IBM zSystems インフラストラクチャー監視ソリューションのイベントとトポロジを、IBM のハイブリッド AIOps プラットフォームである IBM Cloud Pak for Watson AIOps を使用して統合できます。IBM Cloud Pak for Watson AIOps は、複数の機械学習アルゴリズムと高度なイベント相関の機能を適用し、企業が複数のソースからのデータを集約・分析することで問題を特定・解決し、ハイブリッド・アプリケーション全体で根本原因を突き止められるようにします。

図 15 は異常検知のアーキテクチャーの例を示しています。

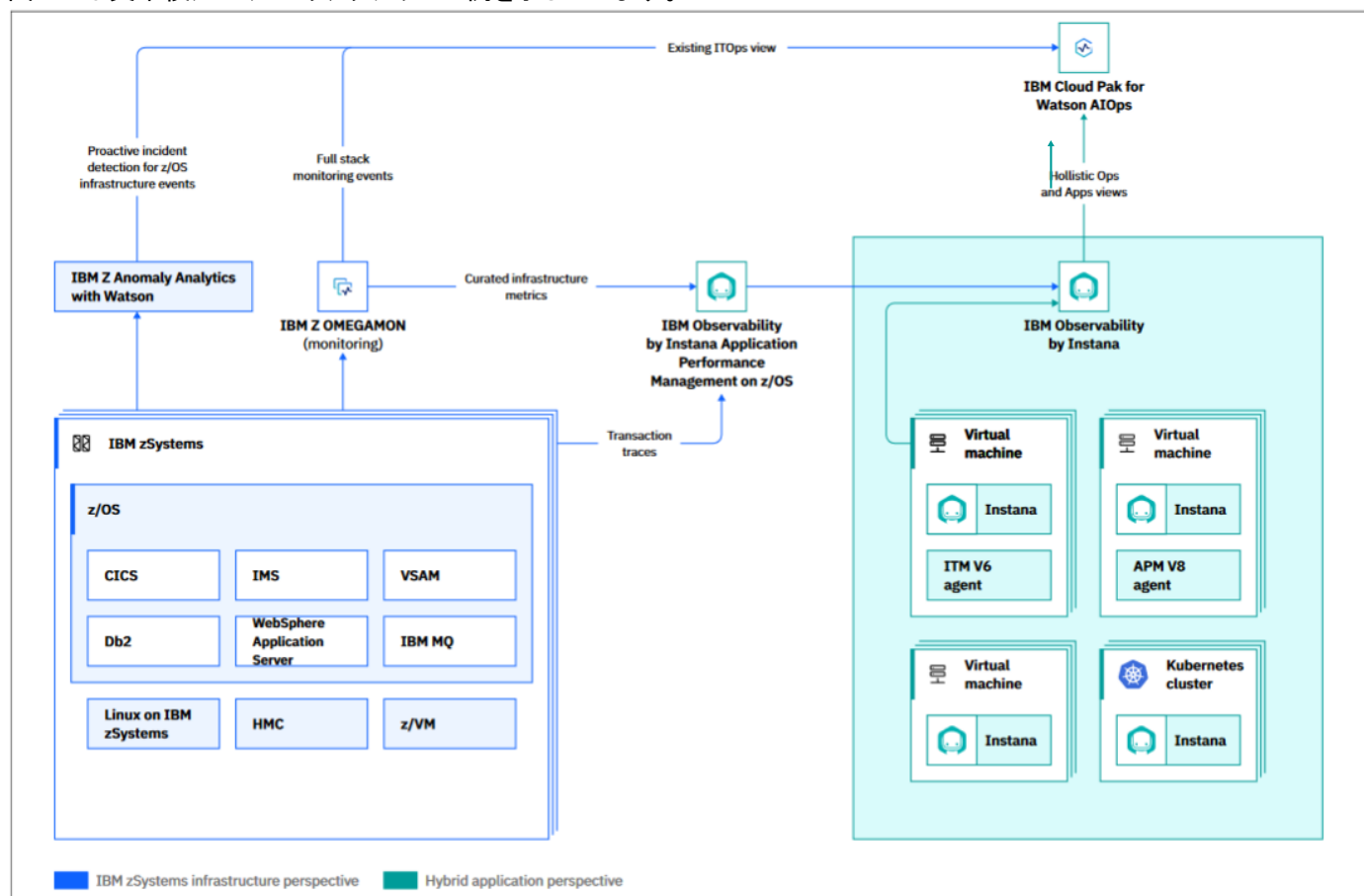


図 15 異常検知のアーキテクチャー例

AI 洞察

2 つ目の原則には、IBM zSystems インフラストラクチャー・チームを AI 洞察で支援し、インシデント検知とアクション実行までの時間を迅速化して、診断時間の短縮に役立てることが含まれます。機械学習と履歴データを活用して、自社の IBM zSystems インフラストラクチャーの通常の動作を示すモデルを構築することにより、IBM zSystems の運用チームは、SLA に影響を及ぼすイベントにつながる可能性のある異常なアクティビティについて事前に警告を受け取ることができます。アラートのしきい値設定の自動化作業を機械に任せることで、従来の監視ソリューションで必要であったしきい値の設定・管理における対象分野の専門家 (SME) への依存を減らすことができます。

IBM zSystems インフラストラクチャー・チームが運用の問題に気が付いたときには、さらなる AI や高度な統計ツールを活用し、トリアージと根本原因分析プロセスを加速できます。

ばらばらな運用上の洞察と IBM zSystems プラットフォームからの洞察を相関することで、ハイブリッド・インシデント識別と診断時間を加速できます。これを行うために、ほぼリアルタイムの運用データを Elastic や Splunk などのエンタープライズ分析プラットフォームにストリーミングして、ハイブリッド運用データを視覚化および検索します。同じように、IBM zSystems インフラストラクチャーの状況をトリアージする場合、履歴データを分析して z/OS サブシステムにまたがる異常の問題のシグニチャーを特定、相関、優先順位付けすることによって、アプリケーション・ワークロードの影響、クリティカルなインフラストラクチャーの状況、および停止の根本原因診断の時間を大幅に短縮できる可能性があります。

自動化されたワークフロー

AIOps の3 つ目の原則は、自動化されたワークフローを通じたインシデント管理の迅速化に関するものです。活用できるツールや手法は複数ありますが、チーム間のコラボレーションや製品間の統合に最も重要なハブとなるのが IBM Z ChatOps です。

IBM Z ChatOps は、IT 運用チームとアプリケーション・チームが協力し合い、組織のサイロを解消できるようにするための会話型インターフェースを提供します。加えて、IBM zSystems インフラストラクチャー・チームが、自らのサービスを利用するアプリケーション・チームとメインフレームの洞察を共有できるようになります。これは、Slack、Microsoft Teams、Mattermost、および IBM zSystems 管理ツールなどのチャット・ツールとの統合を通じて実現します。

この統合により、IBM zSystems インフラストラクチャーの運用担当者と SME は、チャットで IBM zSystems ドメインのツールに問い合わせを行い、洞察を表示できるようになるため、SRE やアプリケーション・チームを含む、トリアージに関与する誰もが完全な透明性を得て、知識の共有を加速できます。

総じて、メインフレームのモダナイゼーション・ジャーニーに乗り出す企業にとって、IT 運用チームが、新しく開発されたアプリケーションや機能のスピードや複雑性をどのように管理して、顧客満足度を確保できるかを検討することが重要です。

IBM zSystems インフラストラクチャーが自社のハイブリッド・アプリケーションをどのように支えるかを理解した後の次のステップは、インフラストラクチャー・タスクを迅速化・自動化するために、ドメイン固有のツールをさらに追加することです。AIOps ソリューションを IBM zSystems インフラストラクチャーの運用と管理に適用する方法についての詳細は、「IBM Z の AIOps へのハイブリッド・アプローチをとるためのベスト・プラクティス ([Best practices for taking a hybrid approach to AIOps on IBM Z](#))」をダウンロードしてください。

VI. AI 主導の大規模な意思決定

ここまで、ビジネスを促進し、メインフレームのプロセスを最適化するのに役立つ各エントリー・ポイントについて説明しました。次のエントリー・ポイントは、IBM zSystems プラットフォームの AI が、アプリケーションやビジネス・トランザクションにインテリジェンスをもたらす方法について扱います。次に挙げるユースケースを含めて説明します。

- ▶ 運用の異常検知
- ▶ z/OS アプリケーション内の不正検知モジュール
- ▶ 効率的でクリーンなデータを提供するデータ分析におけるインテリジェンス

AI はもはや一部の企業に限定された実験ではありません。現在の AI は、データを実用的な洞察と行動に変換し、飛躍的な前進をもたらすイノベーションを実現することによって、人間の能力を増幅し、リスクを軽減し、運用効率を向上させ、総資産利益率を高めることができる大きな機会を提供します。今後も AI は、企業が事業を運営する方法に深い衝撃を与え、ほぼすべてのセクターを変革するでしょう。したがって、AI を導入することは、ビジネス上の必須事項であり、企業に選択の余地はありません。

データは、情報ベースの経済における基軸通貨と言えます。価値のある洞察は、このデータから引き出されます。どの企業にも蓄積してきたビジネス・データがあり、こうしたデータは、結論を導き出す過程において価値のある洞察を提供できます。しかしながら、そこに立ちはだかる課題は、必要な時に必要な場所でこうした洞察を得ることです。言い換えれば、アプリケーションの SLA に影響を与えることなく、企業のミッション・クリティカルなワークロードで AI を使用するにはどうすればよいでしょうか？

理想的には、AI をすべてのトランザクションに適用することで企業は非常に大きな価値を得ることができます。しかし、IBM が Celent 社に依頼した調査報告書「IBM z16 における不正防止の運用: 銀行、カード決済、決済サービス業の損失を軽減 (Operationalizing Fraud Prevention on IBM z16: Reducing Losses in Banking, Cards, and Payments)」では、リアルタイムの AI スクリーニングが行われるのは、大量のエンタープライズ・ワークロードのトランザクションのうち、わずか 10% であることが指摘されています。このように、AI の運用化にはさまざまな課題があります。

IBM zSystems プラットフォームにおける AI 推論

IBM の戦略は、お客様が IBM zSystems プラットフォーム上の自社のアプリケーションにリアルタイムで AI を融合するためのハードウェアとソフトウェア機能を提供することです。そのため、IBM z16 のプロセッサは、高速でレイテンシーが最適化された推論を目的として設計された、業界初のオンチップ AI アクセラレーターを搭載しています。IBM z16 のプロセッサは、1 ミリ秒の応答時間で、1 日あたり 3000 億回の推論操作を処理できます。また、最も厳しいアプリケーション SLA を維持しながら、IBM zSystems ワークロードに合わせて拡張し、すべてのトランザクションをスコアリングできる最適化された推論を使用して、一貫した応答時間を提供します。

IBM は、Open Neural Network Exchange (ONNX) や IBM Deep Learning Compiler (DLC) などのオープンソース・テクノロジーに投資しています。また、z16 アクセラレーターを土台に、人気の高いオープンソース・データサイエンス・テクノロジーをポーティングして AI エコシステムを構築しています。企業は、このインフラストラクチャーを活用して、IBM zSystems プラットフォーム上で任意のソフトウェアを使用できます。

AI エコシステムは、TensorFlow のような多くのオープンソース AI フレームワークを活用できるように最適化されており、オープンソース製品を IBM zSystems プラットフォームにシームレスに統合することができます。このインフラストラクチャーは、オンプレミス、パブリッククラウド、プライベートクラウド、ハイブリッドクラウドなどのあらゆるプラットフォーム上のあらゆるフレームワークでの AI モデルの構築とトレーニング、および IBM zSystems プラットフォームへの AI モデルのデプロイメントにおける柔軟性を促進します。

AI を IBM zSystems プラットフォーム上のデータやコア・ビジネス・アプリケーションと緊密に統合することにより、IBM zSystems プラットフォームのサービス品質を享受でき、アプリケーションと AI のレジリエンス、99.99999% の可用性、AI 資産の可搬性、および IBM zSystems が誇る企業データの安全性がもたらされます。

図 16 は、IBM z16 でのデータの編成、トレーニング、およびデプロイメントの仕組みを示しています。

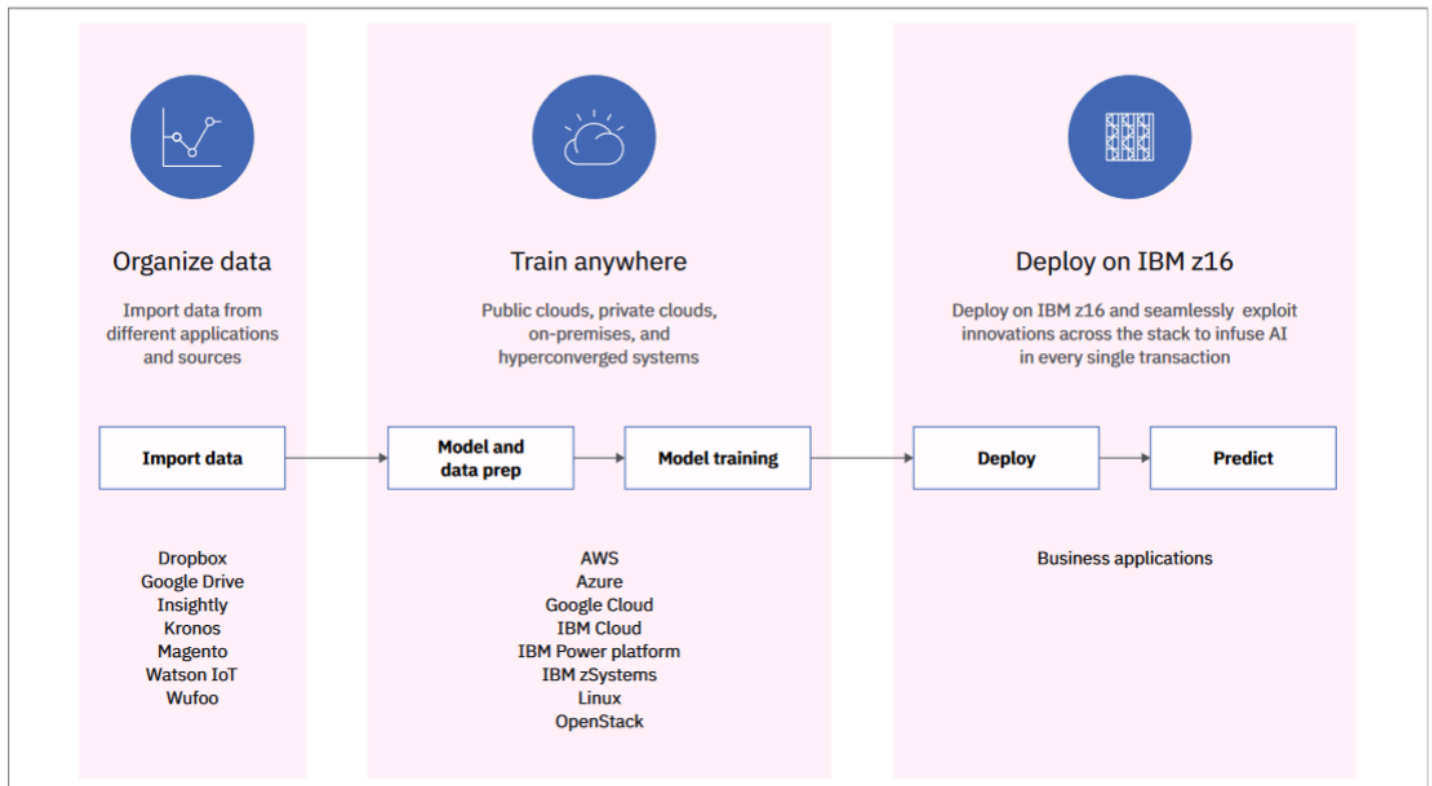


図 16 IBM zSystems プラットフォーム上の AI 推論

AI 推論の導入による z/OS アプリケーションの強化

次に挙げるポイントは、企業が自社 z/OS アプリケーションに AI 推論を導入して強化することに関心を寄せている理由の一部を説明するものです。

▶ AI が与える多くの業界への影響。

金融セクターだけでも、クレジットカード取引拒否により、2980 億ドルの回避可能な損失があったと推定されています。¹⁰ 加えて、銀行が既存の不正検知アルゴリズムでスコアリングできるトランザクションはわずか 10% であり、その結果として大きな収益損失が発生しています。

▶ オンプレミスの不正検知。

例えば、ある米国の銀行では、すべての取引をリアルタイムでスコアリングして不正パターンを検知し、不正なトランザクションを防止することができませんでした。オフ・プラットフォームの AI 推論ソリューションを使用していたときには、SLA への影響が理由でトランザクションの 80% をスコアリングできないでいました。しかしながら、IBM z16 のオンチップ AI アクセラレーターを採用することで、この米国の銀行は、今やトランザクションの 100% をリアルタイムかつオン・プラットフォームでスコアリングし、SLA 内の応答時間を維持できるようになりました。その結果、大きなコスト削減と顧客満足度の向上が実現しました。

▶ 清算と決済。

不正なトランザクションを防止することは、信頼を維持し、顧客体験を向上するための最重要事項です。あるクレジットカード処理企業は、決済の前に、リスク・エクスポージャーが高く、外貨が関与している取引やトランザクションを予測する必要があります。これは、SLA とバッチ・プロセス・ウィンドウに影響を与えることなく実現しなければなりませんでした。この企業は、既存のルール・ベースのアプローチを強化するとともに、IBM Integrated Accelerator for AI での高パフォーマンスと低遅延のスコアリングを可能にする TensorFlow を使用した AI ソリューションを埋め込むことで、高額な費用を招く結果と規制違反を回避しました。

¹⁰ <https://www.ibm.com/downloads/cas/DOXY3Q94>

モデルのトレーニングに対する IBM のアプローチ

IBM の AI モデルの開発とトレーニングへのアプローチは、企業がどこでもモデルを構築・トレーニングできるようにし、IBM zSystems プラットフォームにデプロイできるようにすることです。そのため IBM は、あらゆる人気の高いフレームワークを使用して作成されたアセットを IBM zSystems プラットフォームに移植できるようにすることに重点を置いています。この目標は、ONNX テクノロジーを活用することにより達成されます。このテクノロジーを使用することで、データサイエンティストは、任意のフレームワークでモデルを構築し、そのモデルを共通の移植可能な形式に変換できるようになります。ONNX モデルは、IBM Research® が開発した ONNX モデル・コンパイラを使用して、IBM zSystems プラットフォーム上にデプロイできます。このコンパイラは、ONNX-Multi-Level Intermediate Representation (MLIR) プロジェクトを基盤に構築されており、アプリケーションに組み込むことができる高度に最適化された推論プログラムを作成します。

IBM zSystems プラットフォームは、TensorFlow、Spark、scikit-learn などの他の人気の高いオープンソース・データサイエンス・パッケージも使用することが可能で、最適化されたライブラリーや IBM zSystems アーキテクチャーを活用する AI フレームワークとランタイムのコンパイラも装備しています。企業は、Red Hat OpenShift on IBM zSystems とさまざまなオープンソース・パッケージを中心に据えて構築されたクラウドネイティブ開発エコシステムを利用して、IBM zSystems プラットフォームに AI エコシステムを構築することができます。

IBM Watson Machine Learning for z/OS は、IBM zSystems プラットフォーム用の AI と機械学習のプラットフォームです。あらゆるプラットフォームのあらゆるフレームワークで AI モデルをトレーニングし、それらのモデルを IBM zSystems プラットフォームに迅速にデプロイして、トランザクション実行中にリアルタイムでスコアリングおよび推論を行えます。

図 17 は AI エコシステムを構築するために活用できる AI サービスを示しています。

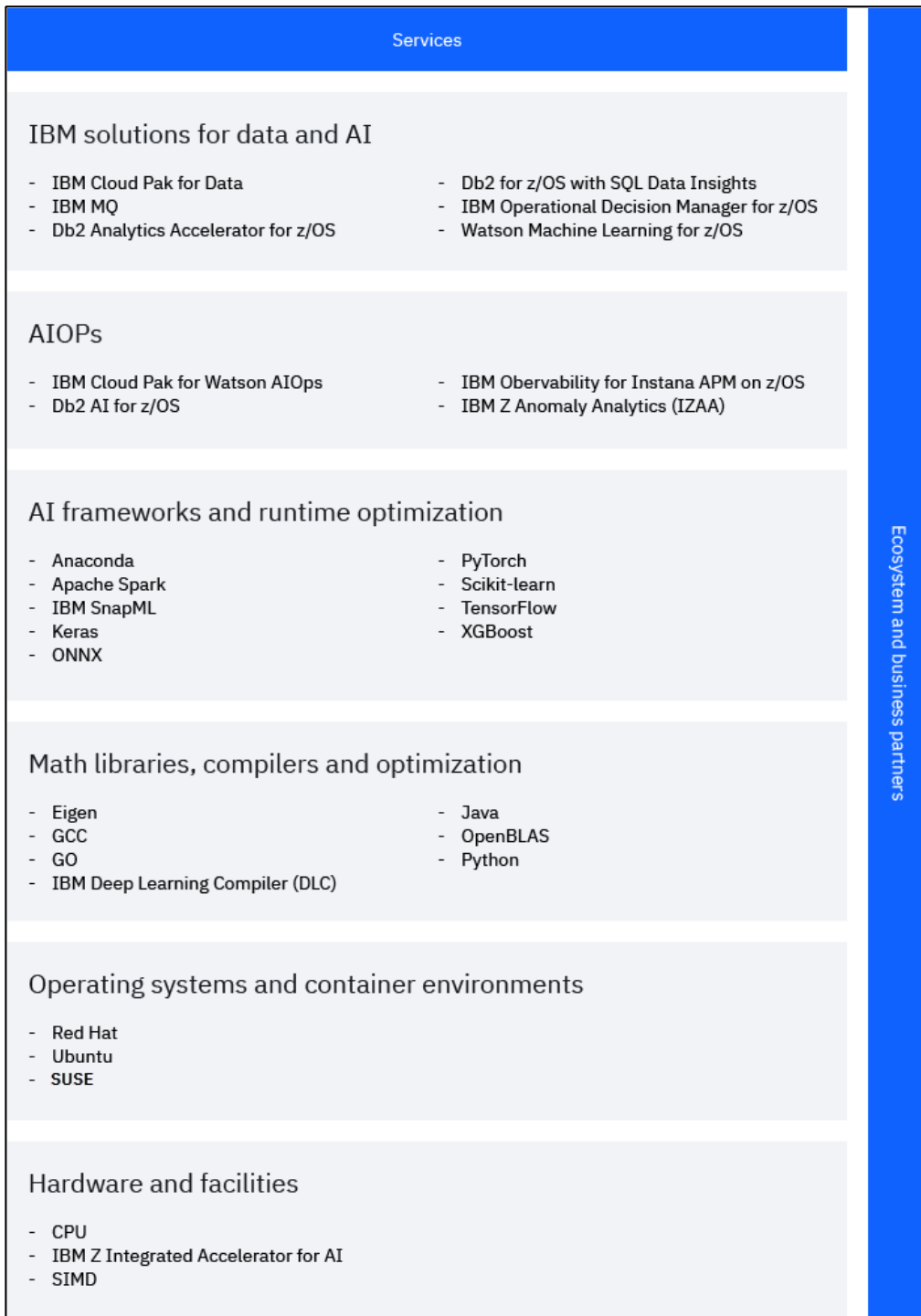


図 17 AI サービス、エコシステム、ビジネス・パートナー企業

AI の導入による高い成果は、さまざまな分野で実証されています。工業セクターにおいては、AI の展開により、効率を改善し、顧客維持率を向上し、生産性を高めて、データ処理機能を大幅に拡張し、人間の知性を補強できます。

VII. IT の自動化と標準化

IT の自動化と標準化も、IBM がベスト・プラクティスとして推奨する継続的なアプリケーション・モダナイゼーション・ジャーニーのエントリー・ポイントであり、何十年にもわたって大規模企業において重要な役割を果たしてきました。

自動化は、ハイブリッドクラウドを導入する大規模企業にとって今や不可欠です。自動化によりビジネス・プロセスと IT ワークフローをコード化することにより、企業は効率的に機能できるようになります。このアプローチを自動化された運用とプロビジョニングと組み合わせることで、企業が必要とする、迅速な日常業務と技術運用を実現することができます。自動化への投資はたとえ小規模であっても、高度なスキルを持つ人員が解放され、モダナイゼーションが加速されることで、顕著な節減が実現できます。

自動化を導入するにあたって限定的なアプローチをとってしまうと、複数のドメインの専門知識が必要になったり、チームや組織で二度手間が発生したり、部門間にサイロが生まれたりすることがよくあります。アプリケーション・モダナイゼーションを進めるにあたって、企業は高まる複雑性に対処し、利用者の期待に応えることができる、堅固で効率的な IT 自動化を実装しなければなりません。この取り組みには、IT インフラストラクチャーの末端に位置するエッジから企業の中央部を形成する IBM zSystems プラットフォームに至るまで、一貫性のある標準化されたアプローチが求められます。

Red Hat Ansible Automation Platform は、複数のクラウド上で稼働でき、組織とさまざまな IT 環境をまたいで自動化を構築・運用するための基盤です。このプラットフォームは、複数のクラウドと IBM zSystems プラットフォームにまたがる企業全体での自動化を実装するために必要なさまざまな機能が含まれています。Red Hat Ansible Certified Content for IBM Z¹¹ には、IBM zSystems 固有の知識をインターフェースにコード化することによって、生産性を向上し、z/OS、z/OS ミドルウェア製品、およびその他の自動化ソフトウェアの自動化を加速するさまざまなコレクションが含まれています。

大規模な標準化

拡大し続ける企業のニーズに対応するために欠かせない IT 自動化ソリューションが、拡張性が高く、長く使えるものであることを確実にするには、企業全体での標準化されたアプローチが必要です。大規模に標準化するために必要なのは、目的、ツール、方法論だけではなく、次のことを達成できるプラットフォームです。

- ▶ 各種ドメイン (DevOps、ネットワーキング、セキュリティ、開発など) わたって拡張し、それらを連携させることができる、幅広い自動化基盤を提供する。
- ▶ オープンであり、統合を通して各ドメインの固有のニーズを満たすことができる。
- ▶ プロセスの融合と合理化を促進する。
- ▶ 迅速な導入、維持、拡張が容易かつ高い信頼性で行える。

Red Hat Ansible Automation Platform

Red Hat Ansible Automation Platform は、デプロイメント・オプションと自動化ソリューションの両方で、ハイブリッドクラウド環境をサポートするように設計されています。Red Hat Ansible Automation Platform を使用したドメイン間のオーケストレーションは、サイロを解消し、ドメインをまたがるツールやプロセスの統合を実現します。それにより、ツールとプロセスの一貫性のあるガバナンスが確立できます。わかりやすい共通のインターフェースが、自動化リソースを活用し、新しい自動化を実装または拡張するためのフレームワークを提供します。

IBM zSystems プラットフォーム上の自動化は、大まかに次の 3 つのカテゴリーに分類できます。

- ▶ 新規システムの構築やユーザーのオンボーディング/オフボーディングなどの、日常の定型タスクのためのユーティリティを備えたシステム・プログラマーのツールボックス。
- ▶ ミドルウェアおよび依存するアプリケーションすべてを含む、システムの完全に自動化された初期プログラム・ロード (IPL) とシャットダウンに関する運用の自動化。これらの運用は、あらゆる種類の復旧にも対応します。例えば、元の場合での再始動や別のシステムへのフェイルオーバーに加え、IBM Geographically Dispersed Parallel Sysplex (GDPS) と組み合わせることで、ローカルやリモート地域での災害復旧にも対応します。
- ▶ 1 日あたり何百万件ものジョブに及ぶ、IBM zSystems プラットフォーム上でのバッチ・ワークロードの実行という広範囲な領域に対応するワークロード自動化。

¹¹ 詳しくは、<https://www.ansible.com/integrations/infrastructure/ibm-zo> を参照してください。

Red Hat Ansible Automation Platform は、さまざまなプラットフォームとインフラストラクチャーにわたる自動化のためのネイティブ・プログラミング・インターフェースを提供しますが、さらなる自動化と統合が必要になるシナリオもあるかもしれません。次に例を示します。

- ▶ **自動化が他の言語で記述されている。**この場合、自動化は Red Hat Ansible で実行することも、個別に実行することもできます。最新の自動化言語は、ハイブリッドクラウドの自動化戦略に適した応用性、柔軟性、およびスキルを提供します。

Python、Groovy、および Java などは、これらの基準を満たす言語の一例であり、特に IBM がサポートする、メインフレーム用に開発されたディストリビューションがあるため、メインフレーム上で最新のアプリケーションや自動化を開発できるスキルを持つ開発者の層が広がります。Python は、今日の自動化の業界標準であり、IBM Z and Cloud Modernization Stack を導入することで、IBM zSystems プラットフォーム上でサポートも合わせて Python を使用することができます。

- ▶ JCL、REXX、CLIST、z/OSMF ワークフロー、IBM Z System Automation などのネイティブ言語やツールを活用した、**あらゆる既存のメインフレーム自動化との統合。**統合は、Red Hat Ansible Certified Content for IBM Z のコレクションによって提供される機能を通じて行うことができます。

Red Hat Ansible Certified Content for IBM Z には、z/OS やその他の IBM zSystems ミドルウェア製品 (IMS、CICS、IBM Z System Automation、IBM z/OS Management Facility、IBM zSystems Hardware Management Console など) の自動化に役立つモジュール、ロール、およびプラグインを提供するさまざまなコレクションが含まれています。これらのコレクションを使用して、幅広い範囲のユースケースを自動化できます。

自動運用に対して最適化されている [IBM Z System Automation](#) などの製品は、専用の Red Hat Ansible コレクションと統合する、強力なポリシー・ベース機能やリアクティブ機能のほか、z/OS 自動化のために特別に設計された機能を提供します。

同様に、バッチ・ジョブのスケジューリングのために最適化されている IBM Z Workload Scheduler などの製品は、何百万件ものジョブの調整、日次スケジュールの計画、実行の監視、エラーからのリカバリーを実行し、バッチ・サービス・レベルが確実に満たされるようにします。これら製品は、ドメイン、チーム、およびユースケースにわたる可視性を備えた Red Hat Ansible エコシステムと連携しているため、そのまま統合して機能を拡張できます。

37 ページの図 18 は、このワークフローと統合がハイブリッドクラウド環境においてどのように機能するかを示しています。

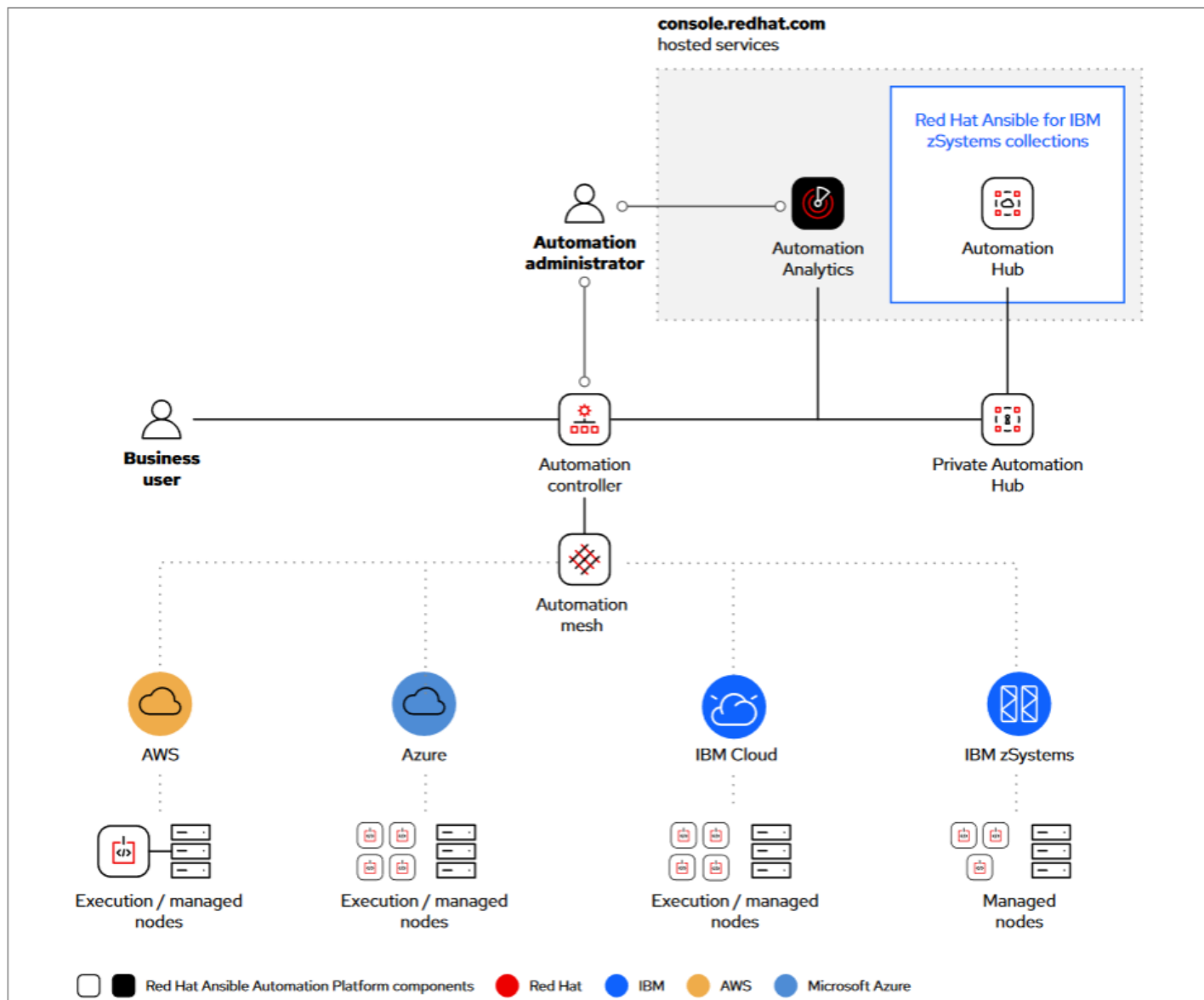


図 18 自動化のリファレンス・ワークフロー

Red Hat Ansible Automation Platform の各種機能、最新の自動化言語、既存の IT 自動化製品、および Red Hat Ansible Certified Content for IBM Z を組み合わせることにより、企業のハイブリッドクラウド自動化戦略に適した統合ソリューションが実現します。これには、次のような例があります。

- ▶ 新しい IMS リージョンが Red Hat Ansible を使用してプロビジョンされる場合、このアクションを IBM Z System Automation に伝達し、IBM Z System Automation によってコントリビュートされるロールを Red Hat Ansible Certified Content for IBM Z に適用することにより、可視性、制御、および自動化された運用を提供できます。このアクションは人手の介入を一切必要とせず、即時に効果を発揮します。既存の利害関係者がこれらのリソースを管理できます。
- ▶ z/OS ジョブ制御言語、z/OS、または TSO コマンドへのアクセスに関する Red Hat Ansible Certified Content for IBM Z の各種機能と同様に、プレイブックから IBM Z NetView または IBM Z System Automation の REST API を活用して、既存の自動化に対し、NetView コマンドの発行やリモート照会、ヘルス・チェックを実行できます。
- ▶ IBM Z System Automation は、z/OS シスプレックスのすべてのリソースの状態を把握できるため、このステートフルな洞察を Red Hat Ansible の自動化に統合することができます。そうすることでシステム状態のディスカバリー・タスクを削減し、正確で最適化された自動化が実現します。例えば、Red Hat Ansible ベースの保守を実行するために、CICS とすべての依存アプリケーションを停止するなどといったことが可能になります。

この他にも、次のような Red Hat Ansible のユースケースがあります。

▶ 構成管理

Red Hat Ansible の宣言型タスク・アプローチと構造的な YAML 構文は Configuration as Code と相乗的に機能します。お客様は、IPL、ネットワーク、セキュリティ構成などのシステム構成をソース・コントロールで管理されたファイルで定義し、Red Hat Ansible のプレイブックで使用することができます。

サーバー構成やプロパティ・ファイルなどのミドルウェアの構成成果物を同じ方法で管理し、ミドルウェアのインスタンスに大規模かつ一貫してデプロイできます。

▶ セキュリティー自動化

z/OS のセキュリティ自動化は、通常、SSL 証明書の更新など、他のプラットフォームから分離された独自のプロセスです。そこで、Red Hat Ansible を使用することで、これらの個別のプロセスを単一の合理的な更新プロセスにまとめ、ターゲットの z/OS システム上のタスクや、オフ・プラットフォームの証明書管理ソフトウェアとのその他のやり取りを連携させた、完全なエンド・ツー・エンドのサイクルを実現できます。

システムの新規ユーザー定義、システム許可機能 (SAF) リソースへの適切な許可とアクセスの提供、またはパスワード・リセットの実行などの、典型的な日常業務を合理化し、z/OS システム・プログラマーのオーバーヘッドを削減できます。¹²

▶ CI/CD とアプリケーションのデプロイメント

Red Hat Ansible を既存の DevOps パイプラインに統合することで、デプロイメント・タスクを迅速化することができます。ミドルウェア・アプリケーションを開発環境およびテスト環境に迅速にデプロイできるとともに、これらのアプリケーションの新しいサブシステム・リソース (新規トランザクション、データベース更新、キュー・リソースの定義など) を作成することも可能です。

▶ オーケストレーション

Red Hat Ansible は、JCL、REXX、z/OSMF ワークフロー、Zowe CLI スクリプト、REST API を使用する既存の自動化やその他の内製の自動化と対話し、オーケストレーションする柔軟性を提供します。

企業全体にわたるより明確な可視性が実現し、ハイブリッドクラウドと IBM zSystems プラットフォームにまたがる既存の自動化をオーケストレーションすることができます。

自動化を新規作成することも、既存の自動化を使用して統合することもできます。例えば、システムや特定のランタイムのステータスについて z/OS の自動化運用ソリューションで状態管理を行い、その状態情報に基づいて、プレイブックを使用して特定のワークロードが実行されている場所 (またはされていない場所) でアクションを実行するといったことが可能です。

▶ システム情報の収集

監査目的で、またはシステム・ステータスやヘルス・チェックのために、システムとセキュリティの構成を収集できます。特定の自動化タスクを実行する前に、システムやリソースに対しプロセス前とプロセス後の検証を実行することもできます。

IBM zSystems プラットフォームを含めるよう企業の IT 自動化の範囲を拡大することが必要です。Red Hat Ansible Platform は、複数のプラットフォームでこのタスクを達成するのに適しています。Red Hat Ansible Platform を使用することで、新規の自動化を作成、または IBM Z System Automation のような企業内の既存の自動化機能を活用して、構成管理、監査やヘルス・チェックのためのシステム・ディスカバリー、プロセス自動化を実現できます。

Red Hat Ansible は、プロビジョニングやデプロイメントの CI/CD インフラストラクチャーとアプリケーション・パイプラインに含めることができます。これにより、Infrastructure as Code (IaC) のような最新の IT 管理プラクティスを使用するその他のプラットフォームと同じような制御された方法で、z/OS ミドルウェアのシステムをプロビジョン、管理、および維持できます。構成は、他のプラットフォームと同様に YAML に格納できます。また、自動化スクリプトをバージョン管理できるため、ガバナンスとコントロールも確保できます。

¹² [System Automation for z/OS: Customizing and Programming](#)

z/OS リソースのセルフサービス管理

IBM Z and Cloud Modernization Stack は、z/OS ミドルウェア環境と z/OS ソフトウェアのインストールのセルフサービス管理を含む、共通の一貫した Platform-as-a-Service (PaaS) 体験を提供します。これらのソフトウェアやミドルウェアのインスタンスは、z/OS を含むハイブリッド・アプリケーションの開発者が作成・管理できます (図 19 参照)。

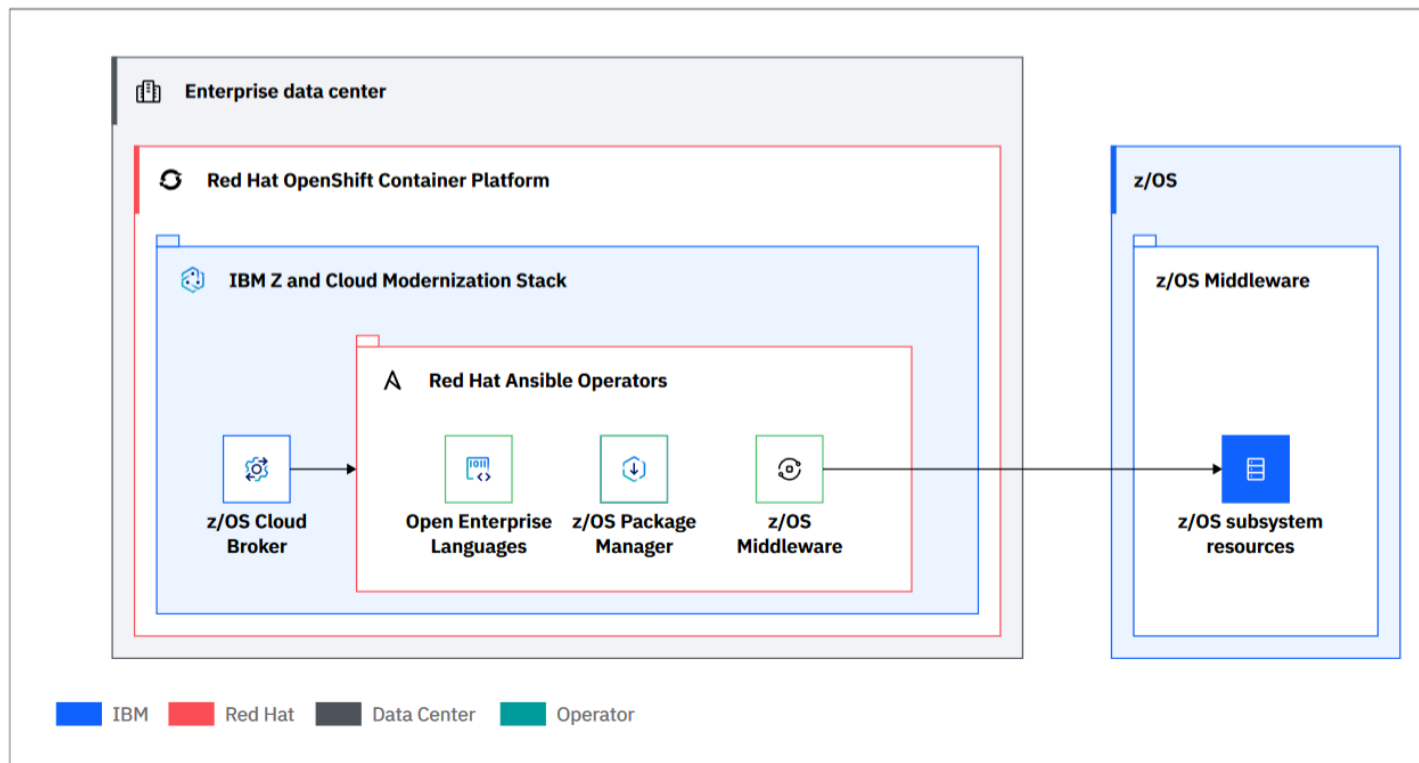


図 19 z/OS ミドルウェアへのセルフサービス・アクセス

ソフトウェアのライフサイクル管理

IBM Z and Cloud Modernization Stack は、IBM Cloud Container Registry に保管され、そこから取得できる Open Container Initiative (OCI) 成果物を使用して、z/OS 対応の言語やツールのインストールを簡素化します。次に挙げるユーティリティと言語は、SMP/E スキルや UNIX ベースのソフトウェアを z/OS にインストールするための一般的ではないファイル形式を使わずにインストールできます。

- ▶ IBM Z Open Automation Utilities
- ▶ IBM Open Enterprise SDK for Python
- ▶ IBM C/C++ for Open Enterprise Languages on z/OS
- ▶ IBM 64-bit SDK for z/OS, Java Technology Edition
- ▶ IBM Open Enterprise SDK for Go
- ▶ IBM Open Enterprise SDK for Node.js

ミドルウェアのライフサイクル管理

IBM Z and Cloud Modernization Stack を使用することで、ユーザーはセルフサービスで z/OS ミドルウェア環境のプロビジョンと管理ができるようになります。ユーザーは、Red Hat OpenShift カタログの利用可能なサービスを通じて z/OS ミドルウェア環境を要求することができます。ミドルウェア環境は、IBM zSystems プラットフォームの論理パーティション (LPAR) または Wazi Sandbox のような仮想化された環境に対してプロビジョニングできます。

z/OS ミドルウェア環境の完全なライフサイクル管理を行うことができ、それには次の項目も含めることができます。

- ▶ 新規環境の構築と撤去
- ▶ プロビジョンされた環境に対するアクション (開始、停止、バインドなど)
- ▶ z/OS にデプロイされたミドルウェア環境間の最適化された統合
- ▶ リソース監視および希望する状態に戻る調整の自動化

IBM Z and Cloud Modernization Stack は、z/OS プラットフォームを活用するために専門的な z/OS スキルを必要としないユビキタスな PaaS 体験を提供します。

モダナイゼーションのリファレンス・アーキテクチャー

IBM zSystems プラットフォームとクラウドを組み合わせたハイブリッド戦略は、スキル・ギャップを大幅に削減し、アジャイルな DevOps アプローチによる迅速なイノベーションを可能にし、大きな変更を加えることなくアプリケーションやデータへのアクセスを容易にし、アプリケーションの運用や拡張のコストを最適化できます。

このアプローチにより、ビジネスの俊敏性と ROI を最大化することができます。これまでのセクションのエントリー・ポイントでは、ハイブリッドクラウドでメインフレーム・アプリケーションをモダナイズするためのさまざまな方法を説明しました。このセクションは、AWS、Microsoft Azure、および IBM Cloud との提携を通じて開発された、具体的なリファレンス・アーキテクチャーを説明します。

- ▶ API を使用したデジタル・チャネルからのアクセスの簡素化。
- ▶ z/OS アプリケーションとクラウド間でのリアルタイムの情報共有。
- ▶ クラウド上の z/OS アプリケーションでの DevOps 活用。
- ▶ IBM zSystems プラットフォームとクラウド全体の IT 自動化。

IBM Cloud

このセクションでは、IBM Cloud を使用したメインフレーム・アプリケーション・モダナイゼーションのソリューション・アーキテクチャーのパターンを説明します。

API を使用したデジタル・チャネルからのアクセスの簡素化

IBM zSystems プラットフォームから公開された REST API は、IBM Cloud 上で稼働するハイブリッドクラウド・アプリケーションに統合できます。金融サービス検証済みとして指定されている IBM Cloud サービスは、API およびクラウドネイティブ・ビジネス・アプリケーションに必要なオーケストレーション、セキュリティ、運用管理を提供します。

- ▶ [IBM API Connect®](#) は、IBM Z and Cloud Modernization Stack で提供される z/OS Connect を使用して作成される OpenAPI 標準の API を管理・公開できます。
- ▶ クラウド・アプリケーションは、REST API と Kafka をベースに構築された [IBM Event Streams](#) のメッセージングを通じて、[IBM Cloud Code Engine](#) や [Red Hat OpenShift Kubernetes Services \(ROKS\)](#) などのメインフレームに統合されたクラウドネイティブ・ランタイムのサービス内で実行できます。
- ▶ クラウドネイティブ・アプリケーションは、ネットワークが分離された[セキュア・ランディング・ゾーン](#)にデプロイできます。セキュア・ランディング・ゾーンとは、セグメント化されたクローズドネットワークです。NIST などの標準に準拠し、Red Hat OpenShift on IBM Cloud (ROKS) のような規制対象のワークロードのデプロイメントにプライベート・エンドポイントを使用します。

図 20 に示されているリファレンス・アーキテクチャーは、ハイブリッドクラウド環境で IBM zSystems アプリケーションと統合する、IBM Cloud for Financial Services® 仮想プライベートクラウド (VPC) のアーキテクチャー・パターンの全体概要を示しています。

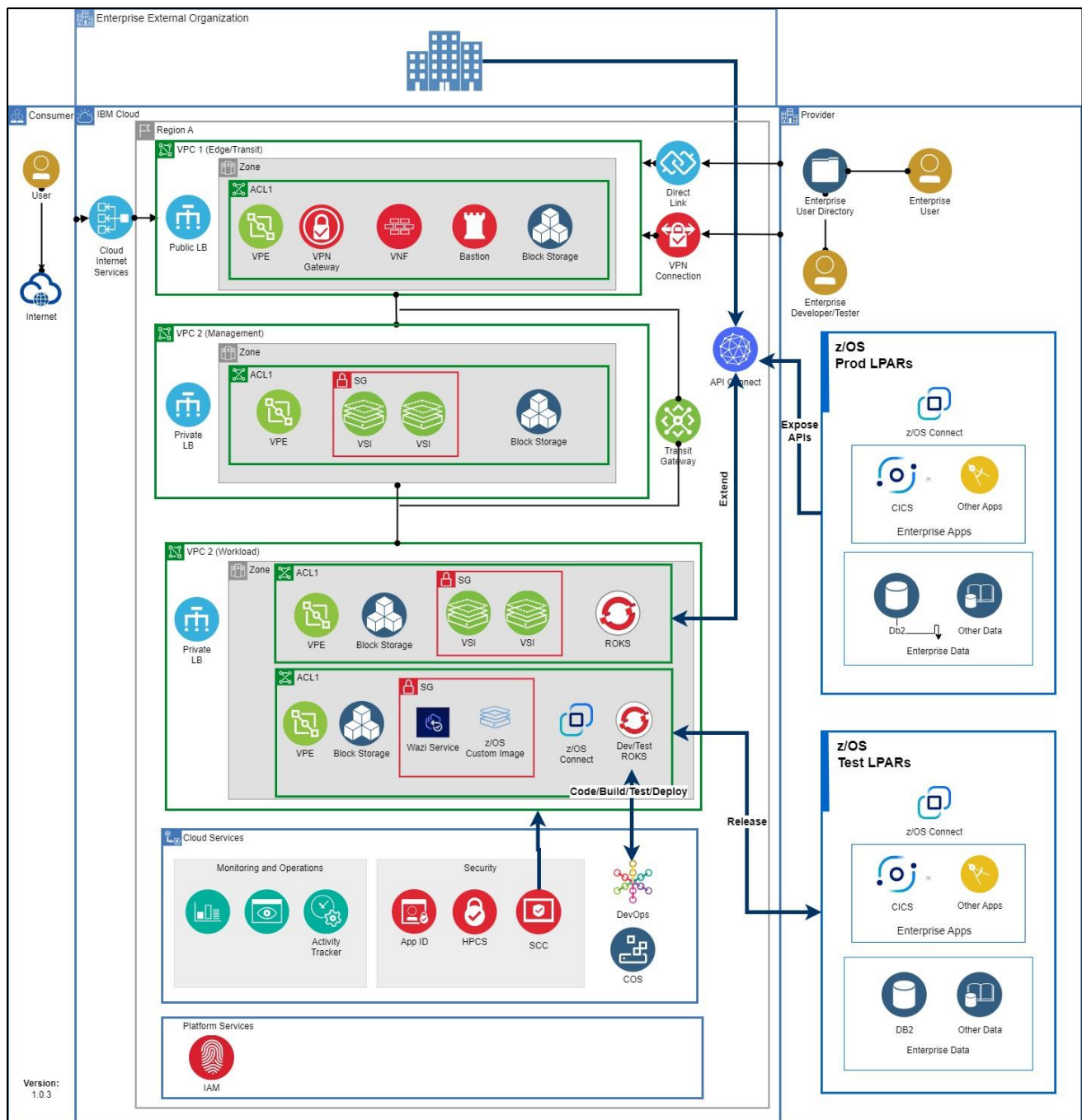


図 20 IBM Cloud for Financial Services VPC アーキテクチャー全体での IBM zSystems API を使用したデジタル・チャンネルのアクセス

このアーキテクチャーは、IBM Cloud のツールチェーンを使って、Infrastructure as Code (IaC) を通じてデプロイできます。IBM Z and Cloud Modernization Stack は開発環境を提供し、IBM Wazi as a Service を使ってプロビジョンされる z/OS 環境と統合します。IBM Hyper Protect Crypto Services (HPCS) および IBM Security® and Compliance Center は、IBM Cloud 内のセキュリティ、暗号化、およびコンプライアンスの各機能を提供します。これらのリソースはすべて、金融サービス検証済み¹³であり、IBM Cloud Framework for Financial Services の一部です。

41 ページの図 20 は、ハイブリッドクラウド環境で IBM zSystems アプリケーションと統合する、IBM Cloud for Financial Services 仮想プライベートクラウド (VPC) アーキテクチャ・パターンの全体概要を示しています。

z/OS アプリケーションとクラウド間でのリアルタイムの情報共有

エンタープライズ・ビジネス・データは、IBM zSystems プラットフォームに存在します。企業はユースケースに応じて、クラウド・アプリケーションからメインフレーム・データにアクセスできるようにするためのさまざまなデータ統合戦略を導入できます。IBM zDIH は z/OS ランタイムと直接統合されたオンプレミスのキャッシング戦略を導入しているため、リアルタイムの情報共有のためのデータを大規模に提供できます。IBM zDIH を使用して公開される情報は、IBM Event Streams を通じて IBM Cloud に統合できます。また、クラウド・アプリケーションやパートナー・エコシステムは、この情報に安全な方法でアクセスできます。IBM Cloud に保管された情報は厳重に機密扱いされ、アクセスできるのはお客様のみです。HPCS は Keep Your Own Key (KYOK: 自分の鍵の保持) 機能を提供するため、鍵階層全体を完全に制御できます。鍵にアクセスできる IBM Cloud 管理者はいません。

図 21 は、IBM zDIH と IBM Cloud を使用したデータ・フローを示しています。

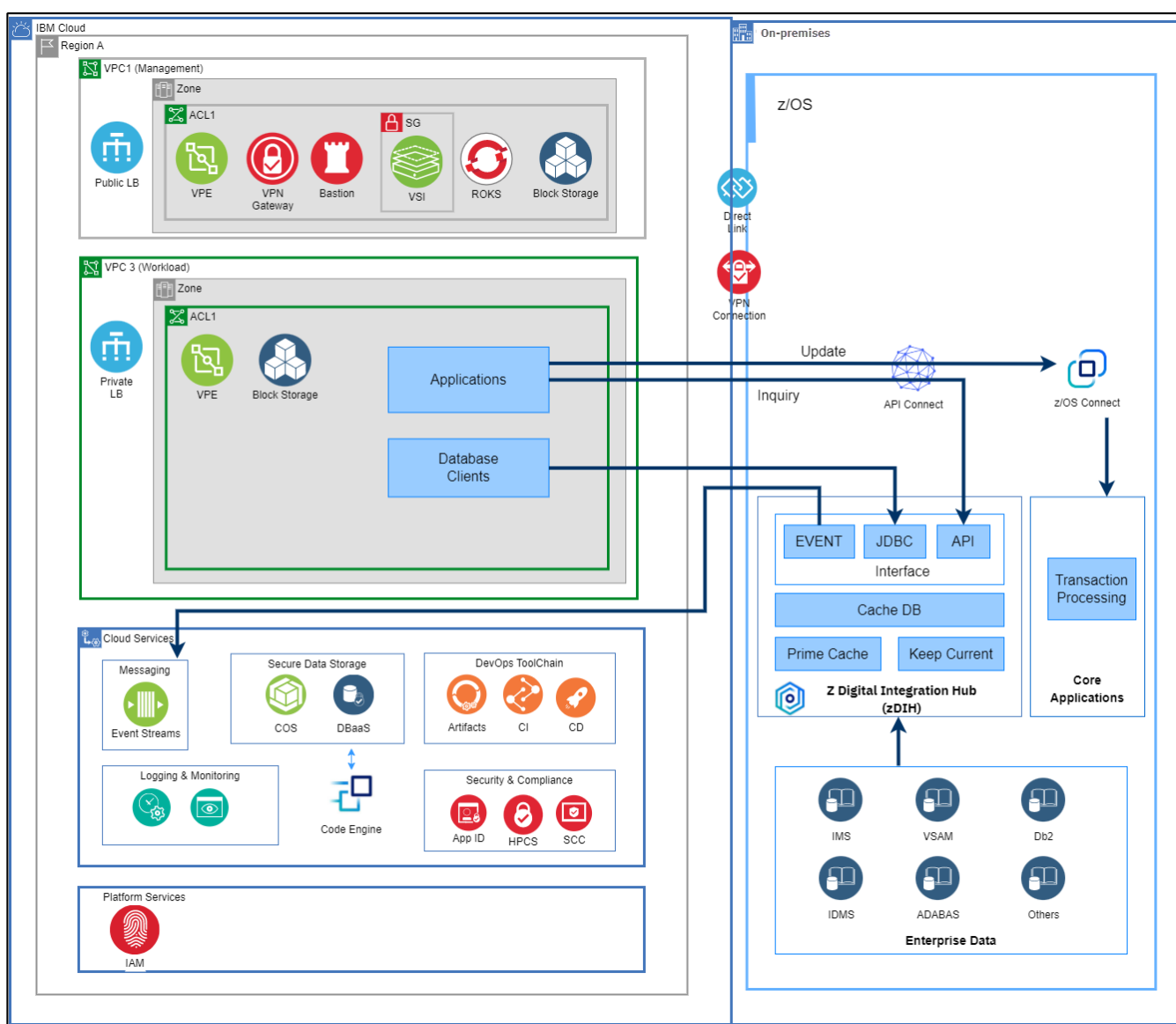


図 21 IBM Z Digital Information Hub を活用したリアルタイムのデータ共有

¹³ 金融サービス検証済み指定されていることは、IBM Cloud Framework for Financial Services の制御要件に準拠していることを意味し、金融機関への価値訴求を強化できる場合があります。

クラウド上の z/OS アプリケーションでの DevOps 活用

IBM z/OS アプリケーションは、IBM Cloud でホストされるエンタープライズ DevOps ツールチェーンに含めることができます。IBM Cloud は、オンデマンドの z/OS 仮想サーバー・インスタンスを提供します。これは IBM zSystems プラットフォームを通じてプロビジョンされ、開発者はセキュアな VPC インフラストラクチャー内で z/OS 開発環境をオンデマンドでスピンアップすることができます。IBM Toolchain および Continuous Delivery は、IBM Cloud やオンプレミスの LPAR 上の z/OS 開発環境と統合できます。そうすることで、z/OS アプリケーションを IBM Cloud 上のサンドボックスで開発し、IBM zSystems の LPAR に安全にデプロイしてテストを自動実行できるようになります。IBM Toolchain および Continuous Delivery には、コード・リスク分析、コード・スキャン、および DevOps Insights による開発サイクルに関する洞察の抽出のためのツールセットが装備され、安全で規制に準拠したパイプラインのための機能を提供します。Secrets Manager および Key Protect は、ゼロトラスト・パイプラインを構築するために必要な暗号化と保護を提供することにより、パイプラインの安全を確保します。

図 22 は、統合された DevOps ツールチェーンを通じた IBM Cloud 内での z/OS アプリケーション開発の概要を示しています。

IBM Cloud 上のエンタープライズ DevOps ソリューション・アーキテクチャーの詳細は、「IBM Cloud によるメインフレーム・アプリケーションのビジネス俊敏性の向上と最新の DevOps プラットフォームの実現 ([Improve business agility and provide modern DevOps platform for mainframe applications with IBM Cloud](#))」を参照してください。

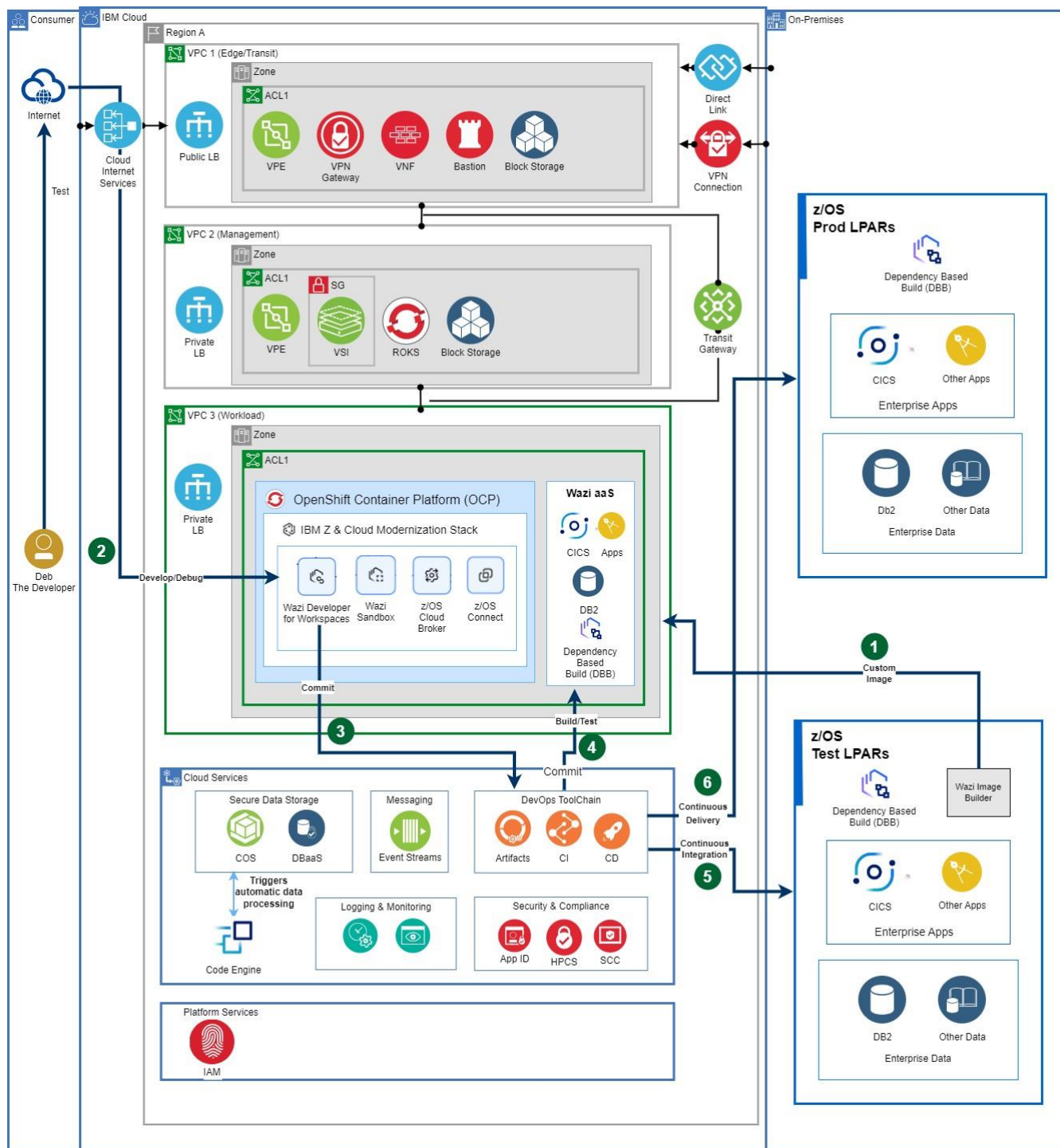


図 22 統合された DevOps ツールチェーンを通じた IBM Cloud での z/OS アプリケーション開発

IBM zSystems プラットフォームとクラウド全体の IT 自動化

非効率の削減、実動コストの低減、IT 環境の複雑性の軽減、およびセキュリティのギャップの特定と解消を実現したいと考えている企業にとって、自動化は最優先事項です。IBM DevOps ツールチェーンと IBM Z and Cloud Modernization Stack の開発環境の機能を活用することによって、IT 自動化のスクリプトを IaC として維持することができます。Red Hat Ansible Platform は、IBM Cloud の VPC にある Red Hat OpenShift クラスター上にデプロイできます。これにより、ハイブリッドクラウド環境の IT 自動化スクリプトをオーケストレーションできます。

図 23 に示されているリファレンス・アーキテクチャーは、IBM Cloud とメインフレームにわたるハイブリッドクラウドでの IT タスク自動化のオーケストレーション・フローを示しています。

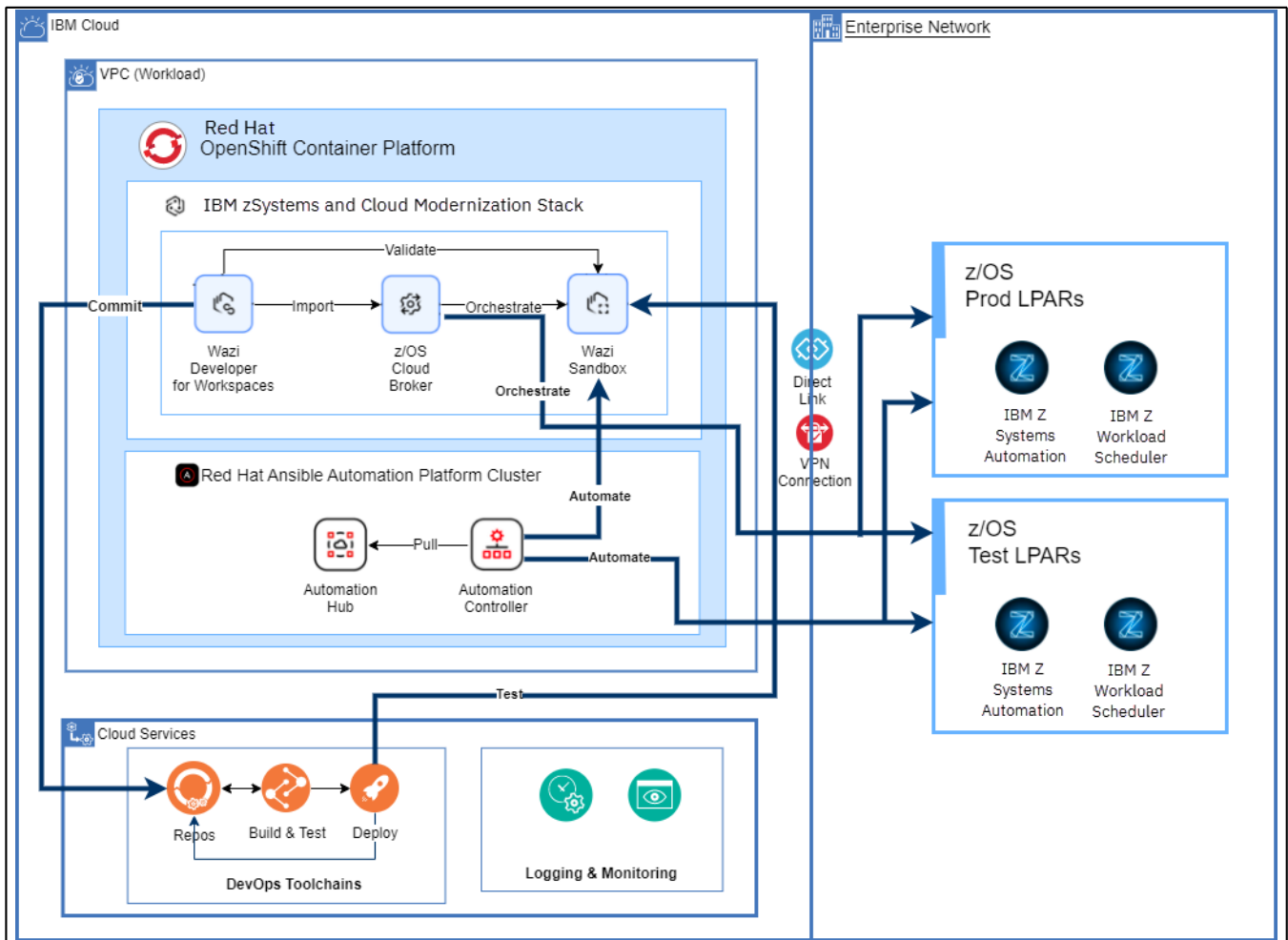


図 23 IBM zSystems プラットフォームと IBM Cloud にわたる IT 自動化

Amazon Web Services

このセクションでは、AWS を使用したメインフレーム・アプリケーション・モダナイゼーションのソリューション・アーキテクチャーのパターンについて説明します。

API を使用したデジタル・チャネルからのアクセスの簡素化

IBM Z and Cloud Modernization Stack のローコード API ソリューション・コンポーネントである z/OS Connect は、AWS 上の Red Hat OpenShift 上でスケーラブルかつセキュアな z/OS API を提供します。この構成は AWS のクラウドネイティブ・アプリケーションが、OpenAPI 標準を使用して RESTful API を通じてメインフレームのアプリケーションとデータに安全にアクセスするためのシンプルな方法を提供します (46 ページの図 24 参照)。

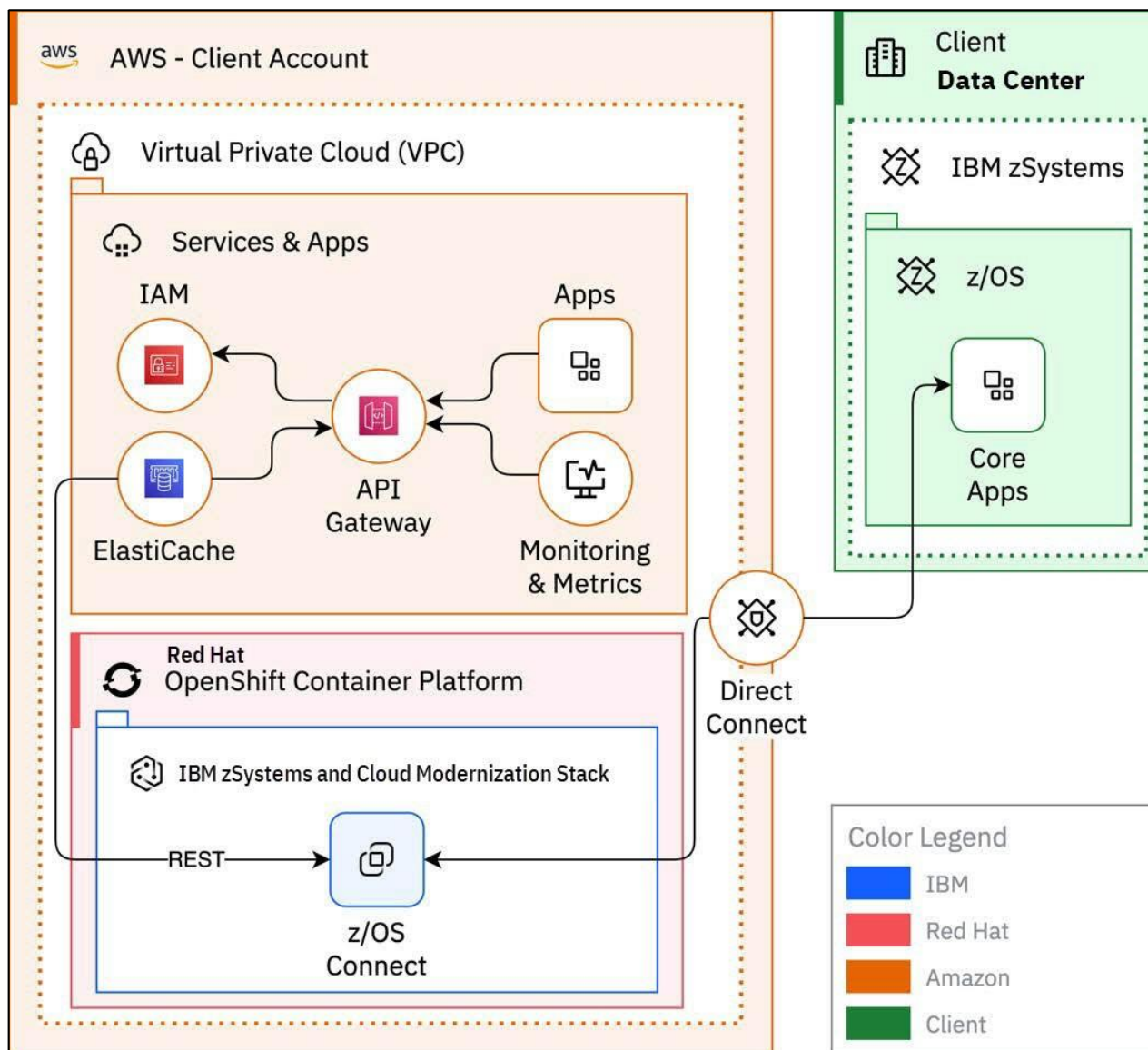


図 24 リファレンス・アーキテクチャー: RESTful API を通じたアクセス

IBM Z and Cloud Modernization Stack と Amazon API Gateway、Amazon ElastiCache、および各種自動スケーリング・サービスが、オンプレミスにあるメインフレームのアプリケーションとデータの低遅延で水平スケーリング可能な統合を提供します。AWS と IBM zSystems プラットフォームの接続には、AWS Direct Connect と仮想プライベート・ネットワーク (VPN) のリンクを使用できます。これにより、低遅延で安全なプラットフォーム接続を確立できます。IBM Z and Cloud Modernization Stack で作成された API は、Amazon API Gateway のようなエンタープライズ API 管理ソリューションと統合できます。Amazon API Gateway は、規模にかかわらず簡単に API の作成、公開、保護を行えるフルマネージド型のサーバーレス・サービスです。AWS 上の Red Hat OpenShift は、ユースケースやチャネルの増加による成長に伴い拡張することが可能です。Web およびモバイル・アプリには、Amazon Cognito を使用できます。ユーザーは Cognito ユーザー・プールまたはフェデレーティッド ID プロバイダー (IdP) を通じてログインできます。

z/OS アプリケーションとクラウド間でのリアルタイムの情報共有

IBM zSystems プラットフォームから大規模に要求され、IBM zDIH を通じて公開されるリアルタイムの情報は、AWS のアプリケーションとサービスに統合できます。AWS のクラウドネイティブ・アプリケーションは、AWS Direct Connect を通じて REST API や JDBC を使用し、このデータに安全にアクセスすることができます。IBM zDIH を通じて公開されるイベントは、Azure クラウド上にインストールされた Kafka メッセージング・ブローカーを通じて、イベント・ストリームにパブリッシュできます。コア・アプリケーションのこのデータのリアルタイム更新は、REST API を使用して実行できます (図 25 参照)。

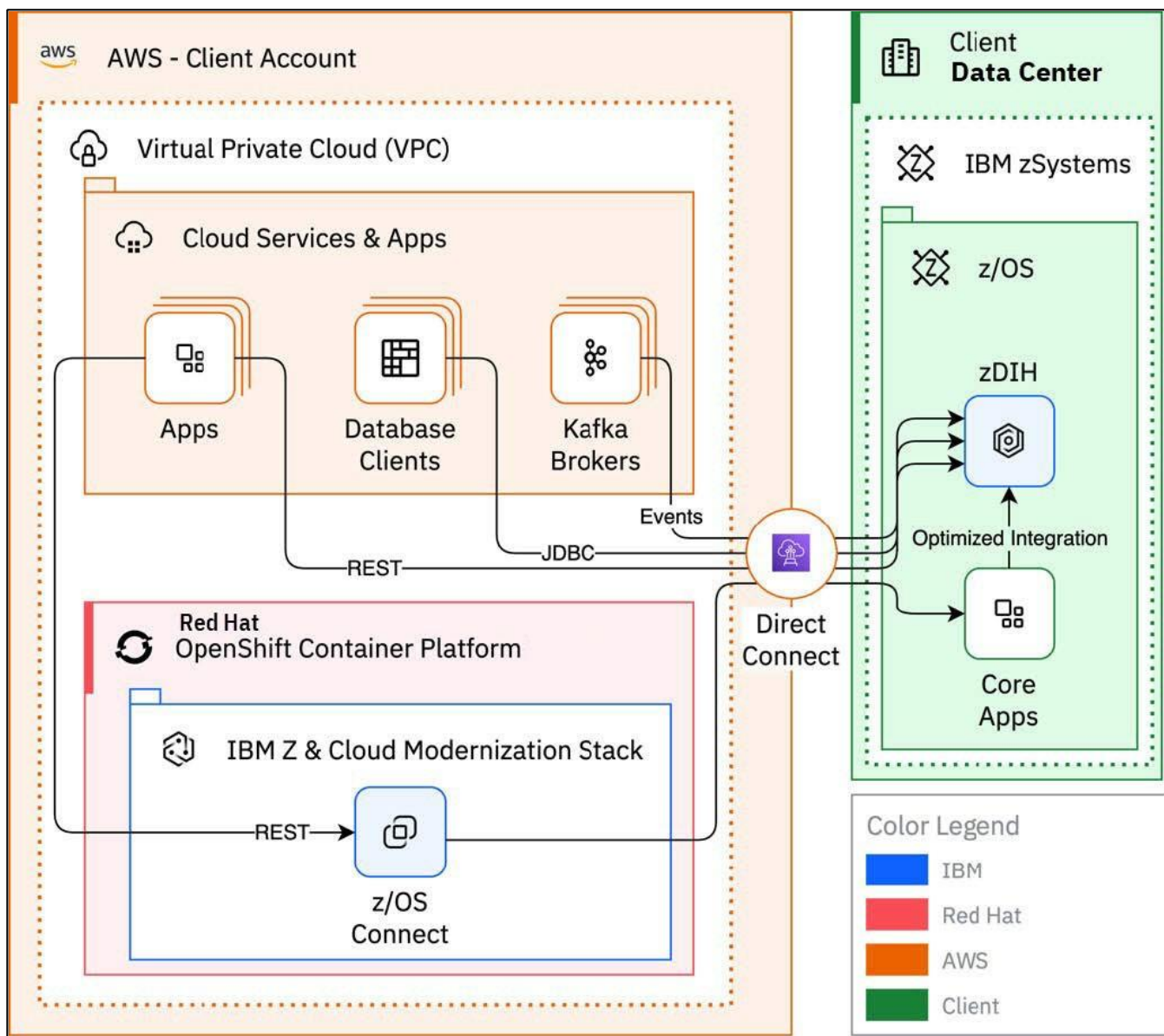


図 25 リファレンス・アーキテクチャー: REST API を通じたアクセス

クラウド上の z/OS アプリケーションでの DevOps 活用

AWS の IBM Z and Cloud Modernization Stack は、z/OS アプリケーションの開発者にクラウドネイティブ体験を提供します。開発や早期テストを Red Hat OpenShift 環境内の z/OS サンドボックスで行うことができるため、チームは分離された環境で開発作業を行うことができます。

48 ページの図 26 は、効果的な DevOps パイプラインのセットアップ方法の例を示しています。

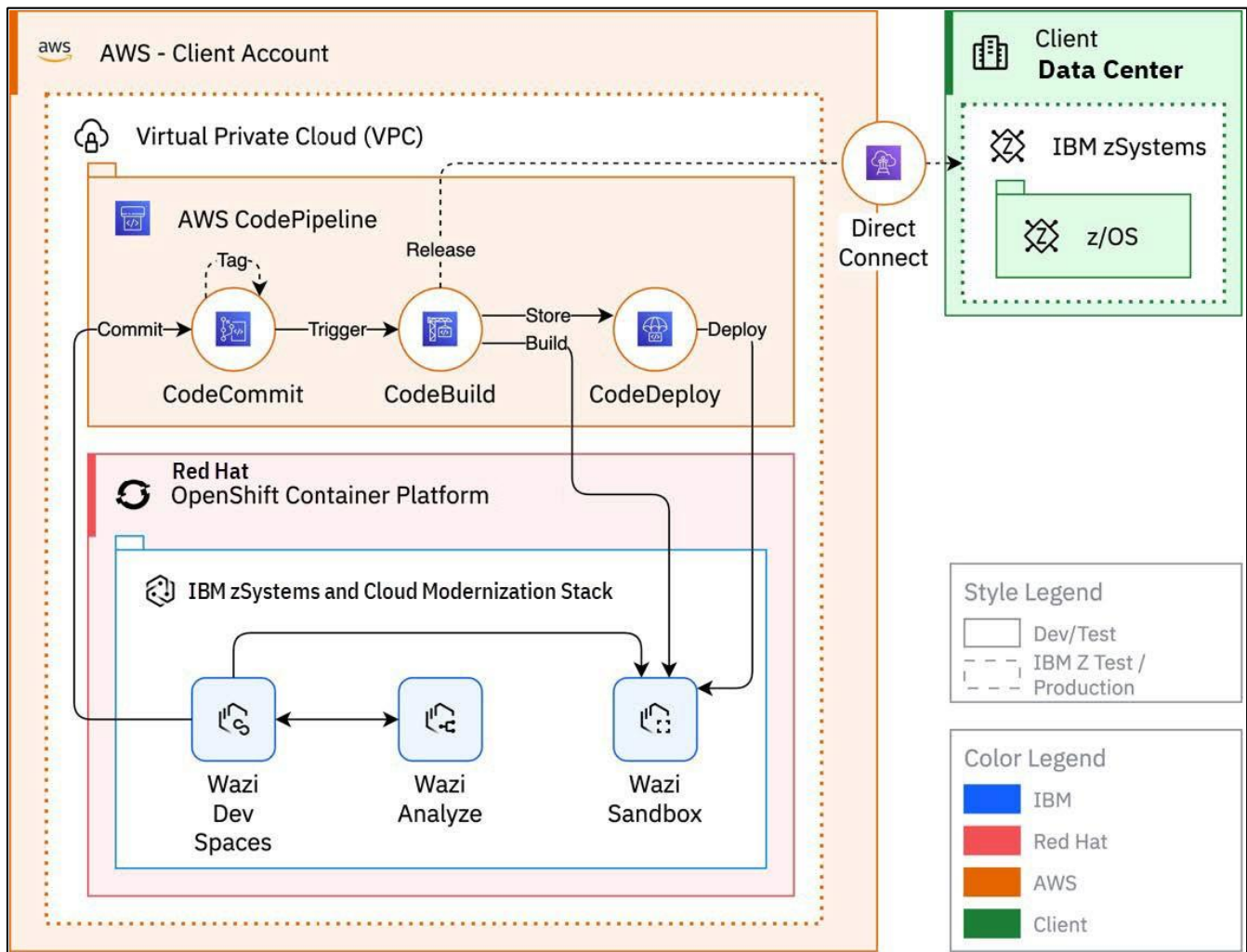


図 26 IBM zSystems アプリケーションの AWS エンタープライズ DevOps パイプライン

IBM Wazi Developer for Workspaces は z/OS アプリケーションをコーディング、ビルド、デバッグするためのブラウザ内統合開発環境 (IDE) を提供しますが、IBM Wazi Analyze と IBM Wazi Developer for Workspaces は、Git などの最新のソース・コントロール管理 (SCM) ツールからソースコードを分析・編集できます。

DevOps ツールは、AWS CodeCommit および AWS CodePipeline と統合されています。AWS CodeBuild や AWS CodeDeploy といった AWS 上のビルド・サービスとデプロイ・サービスは、IBM Z Cloud and Modernization Stack の一部として利用できる z/OS に特化したビルド・サービスとデプロイ・サービスを呼び出します。

Red Hat Ansible は、IBM zSystems プラットフォームをはじめとするあらゆるアーキテクチャーの IT 自動化をサポートします。IT リソースをハイブリッド・マルチクラウド環境全体にわたって一貫性のある方法で自動化し、一元的に管理できるようにすることは、企業にとってメリットがあります。図 27 は、Red Hat Ansible Automation Platform と AWS クラウドの統合アーキテクチャーを示しています。Red Hat Ansible Automation Platform は、サービスとして AWS の VPC にデプロイできます。または AWS 上の Red Hat OpenShift の Kubernetes Operator としてデプロイできます。Red Hat Ansible Automation Platform は、AWS Direct Connect を通じて AWS と IBM zSystems プラットフォームにわたって自動化スクリプトをオーケストレーションできます。自動化スクリプトは、AWS CodePipeline と IBM Z and Cloud Modernization Stack の開発環境の機能を活用することによって、IaC として維持することができます。

The diagram illustrates the Wazi z/OS Cloud Modernization Stack architecture, organized into three main color-coded regions: AWS (orange), Red Hat (pink), and Client (green).

- AWS - Client Account (Orange):**
 - Virtual Private Cloud (VPC):** The central cloud environment.
 - AWS CodePipeline:** Contains **CodeCommit** and **CodeBuild**. A **Commit** triggers a **Trigger** in CodeBuild, which then **Store**s artifacts.
 - Ansible Automation Platform:** Contains **Automation Hub** and **Automation Controller**. The **Automation Controller** **Pull**s artifacts from CodeBuild and **Test**s them.
- Red Hat OpenShift Container Platform (Pink):**
 - IBM zSystems and Cloud Modernization Stack:**
 - Wazi Dev Spaces:** Receives **Import** from Wazi Sandbox and **Validate** artifacts.
 - z/OS Cloud Broker:** **Orchestrate**s between Wazi Dev Spaces and Wazi Sandbox.
 - Wazi Sandbox:** **Automate**s artifacts back to Wazi Dev Spaces.
- Client Data Center (Green):**
 - IBM zSystems:** The target environment, containing **z/OS**.
 - Direct Connect:** A connection point between the AWS VPC and the Client Data Center.
 - Interactions:** The stack **Automate**s and **Orchestrate**s work in the Client Data Center.

Style Legend:

- Solid line: Dev/Test
- Dashed line: IBM Z Test / Production

Color Legend:

- Blue: IBM
- Pink: Red Hat
- Orange: AWS
- Green: Client

49

Microsoft Azure

このセクションは、Microsoft Azure を使用したメインフレーム・アプリケーション・モダナイゼーションのソリューション・アーキテクチャーのパターンを説明します。

API を使用したデジタル・チャネルからのアクセスの簡素化

IBM Z and Cloud Modernization Stack は、z/OS Connect を通じて公開される API を使用して、オンプレミスにあるメインフレームのアプリケーションとデータと Microsoft Azure クラウドのサービスとの、低遅延で水平スケーリング可能な統合を提供します (図 28 参照)。Microsoft Azure ExpressRoute は、クラウドと IBM zSystems プラットフォームを安全に直接接続し、自動スケーリング・サービスは、Redis キャッシュにより提供されます。Azure Virtual Network (VNet) は、Azure 内のプライベート・ネットワークの基本的な構成要素です。

また、IBM Z and Cloud Modernization Stack で作成された API は、Microsoft Azure 上の API 管理ソリューションと統合できるだけでなく、Microsoft Azure の一部として提供される監視サービスやメトリック・サービスを使用します。Microsoft Azure 上の Red Hat OpenShift は、ユースケースやチャネルの増加による成長に伴い拡張することが可能です。

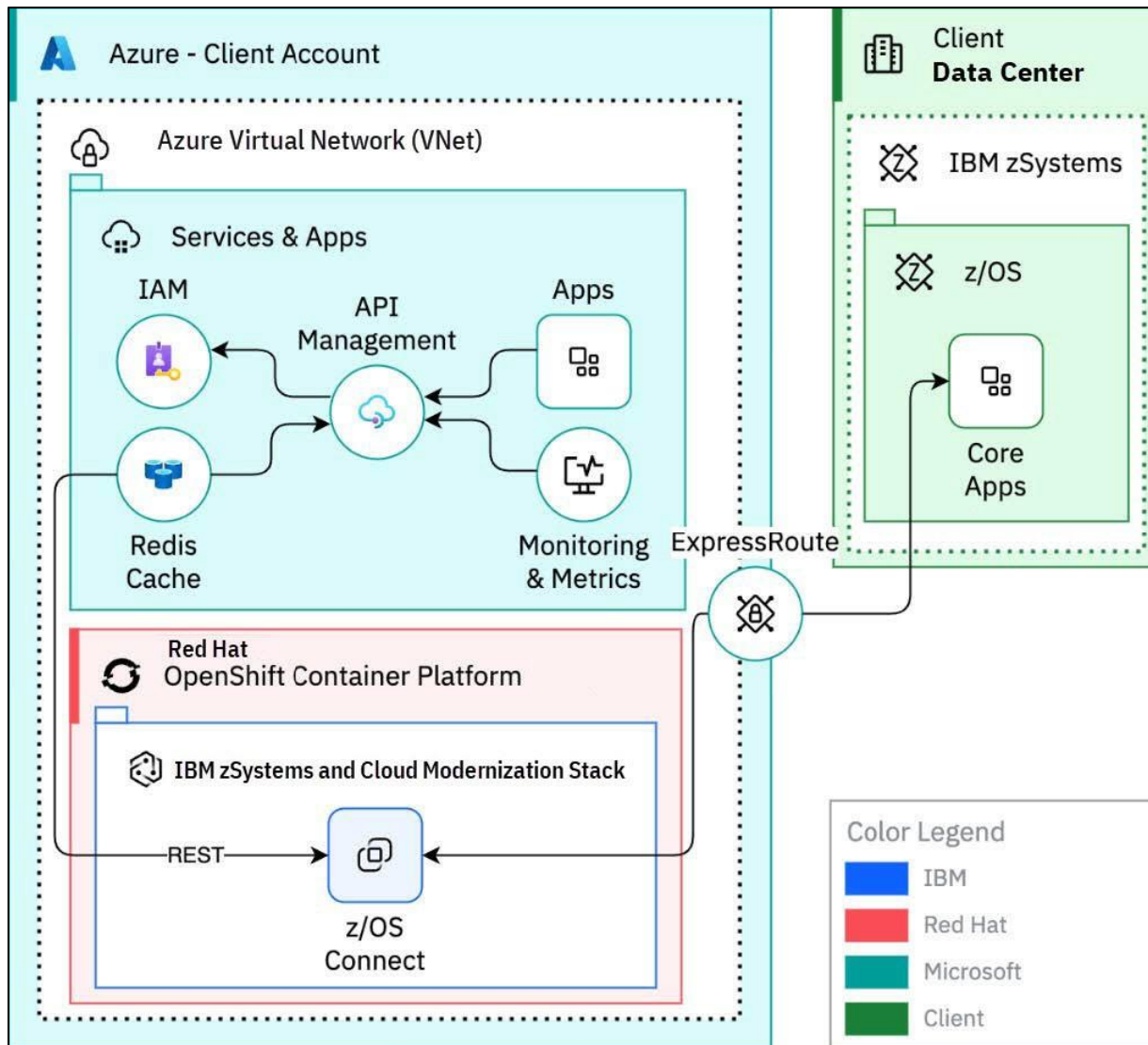


図 28 リファレンス・アーキテクチャー: API を通じたアクセス

z/OS アプリケーションとクラウド間でのリアルタイムの情報共有

IBM zSystems プラットフォームから大規模に要求され、IBM zDIH を通じて公開されるリアルタイムの情報も、Microsoft Azure のアプリケーションとパートナー・サービスに統合できます。Microsoft Azure のクラウドネイティブ・アプリケーションは、Azure ExpressRoute を通じて REST API や JDBC を使用し、このデータに安全にアクセスすることができます (図 29 参照)。IBM zDIH を通じて公開されるイベントは、Azure クラウド上にインストールされた Kafka メッセージング・ブローカーを通じて、イベント・ストリームにパブリッシュできます。コア・アプリケーション内でのこのデータのリアルタイム更新は、IBM Z Cloud and Modernization Stack を通じて公開される REST API を使用することで実行できます。

図 29 は、このアーキテクチャーの例を示しています。

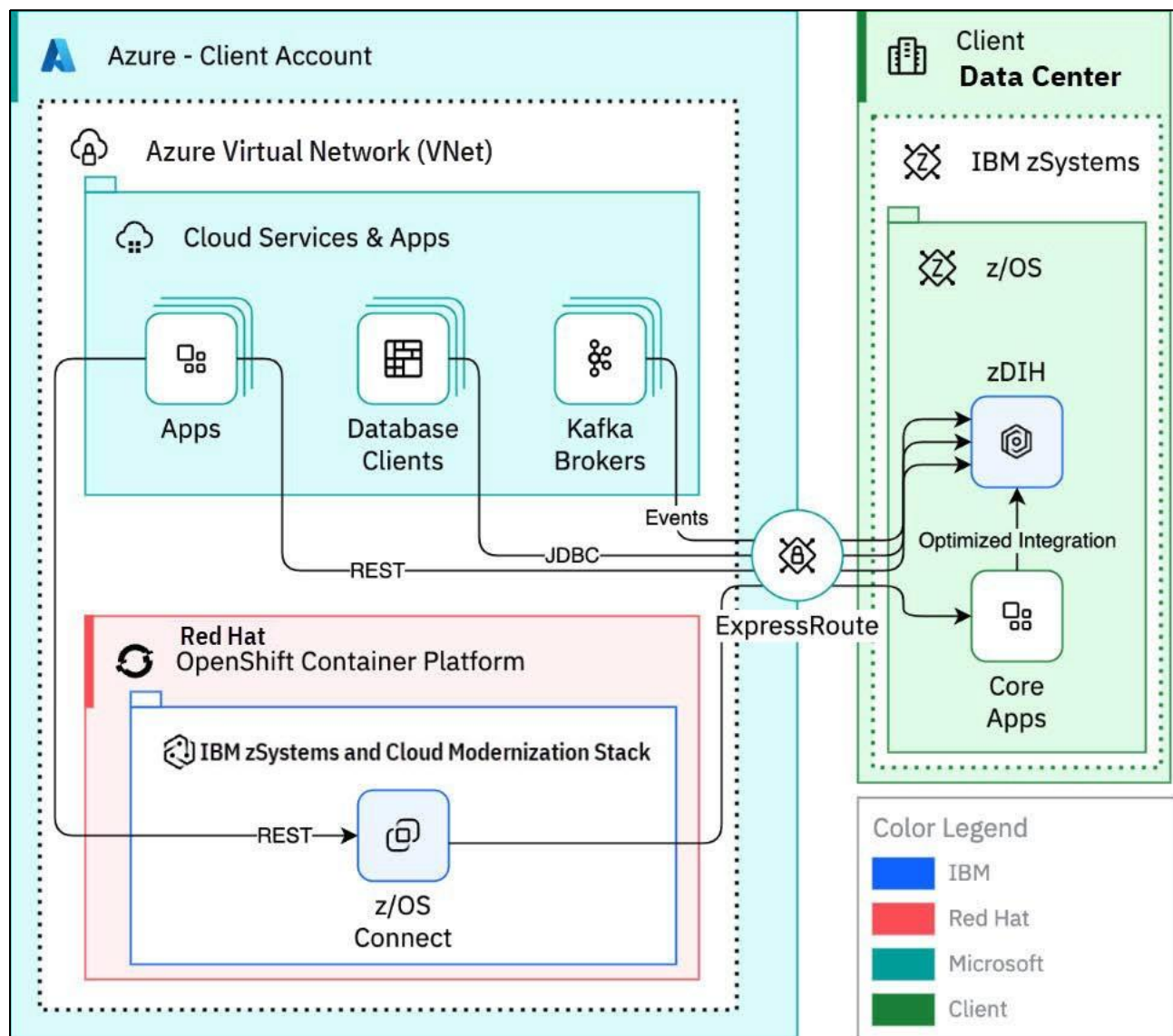


図 29 リファレンス・アーキテクチャー:API を通じたアクセス

クラウド上の z/OS アプリケーションでの DevOps 活用

コーディング、ビルド、デプロイ、テストのための開発環境を作成する IBM のソリューションは、VNet にセットアップされた Microsoft Azure DevOps Services に統合できます。IBM zSystems のアプリケーション・ソースは、企業の他の部分と同じ SCM を使用する Microsoft Azure のリポジトリに保存することができます。

IBM Wazi Developer for Workspacesと IBM Wazi Analyze は、Microsoft Azure のリポジトリにアクセスして、コードの分析、編集、デバッグが行えます。Microsoft Azure パイプラインは、VNet の Red Hat OpenShift にプロビジョンされている Wazi サンドボックス環境に対して実行できます。これは、ビルドを実行し、ビルド成果物を Microsoft Azure Artifacts に保管して、変更を z/OS サンドボックスと IBM zSystems LPAR の両方にデプロイすることによって実現できます (図 30 参照)。

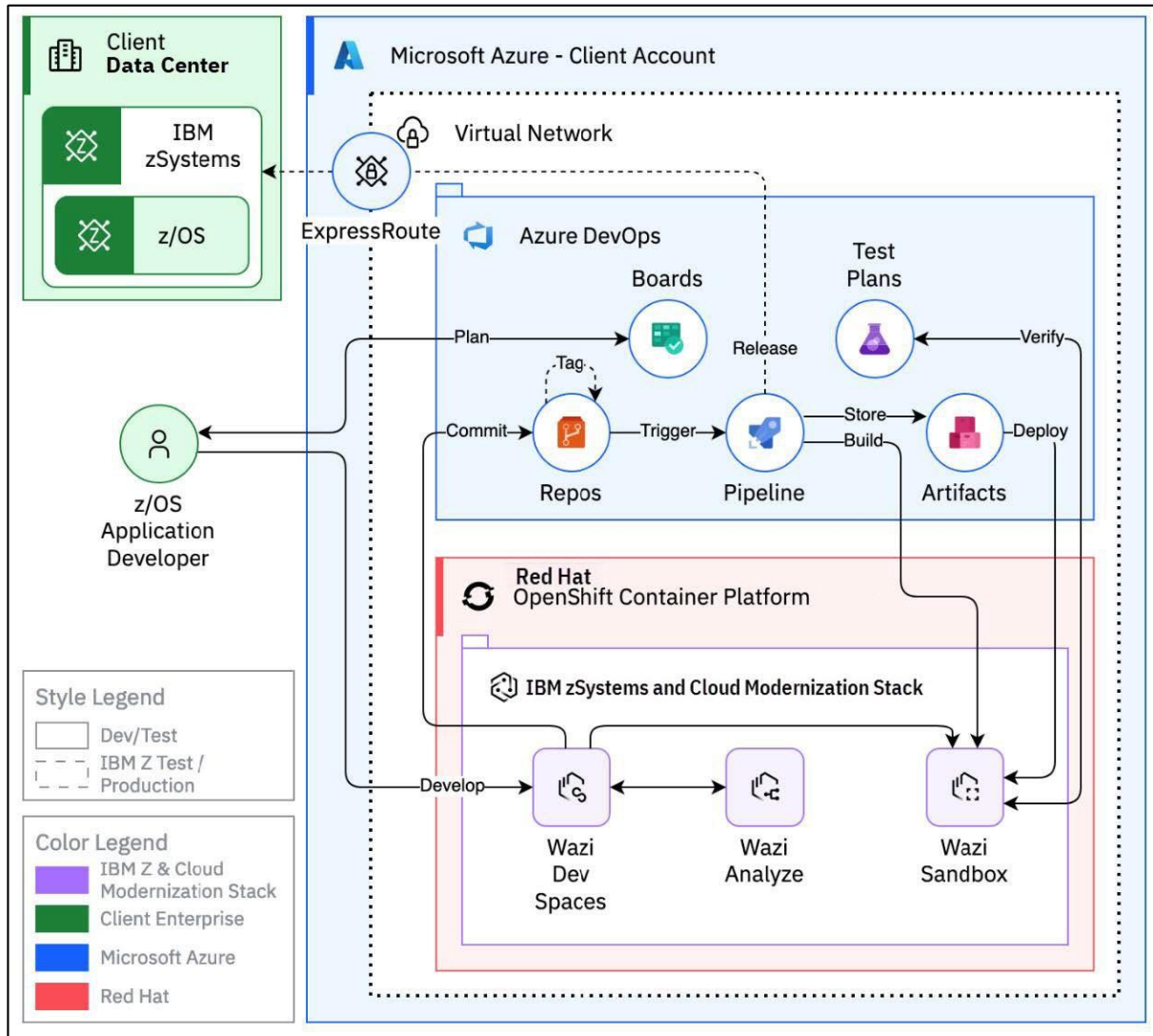


図 30 Microsoft Azure と IBM Wazi Developer for Workspaces および IBM Wazi Analyze

IBM zSystems プラットフォームとクラウド全体の IT 自動化

図 31 は、Red Hat Ansible Automation Platform と Microsoft Azure の統合を示しています。Red Hat Ansible Automation Platform は、Microsoft Azure の VNet にデプロイできます。または、Microsoft Azure 上の Red Hat OpenShift の Kubernetes Operator としてデプロイできます。Red Hat Ansible Automation Platform は、Microsoft Azure ExpressRoute を通じて Microsoft Azure クラウドと IBM zSystems プラットフォームにわたり自動化スクリプトをオーケストレーションできます。自動化スクリプトは、Microsoft Azure DevOps サービスと IBM Z and Cloud Modernization Stack の開発環境の機能を活用することによって、IaC として維持することができます。

図 31 は、このアーキテクチャの例を示しています。

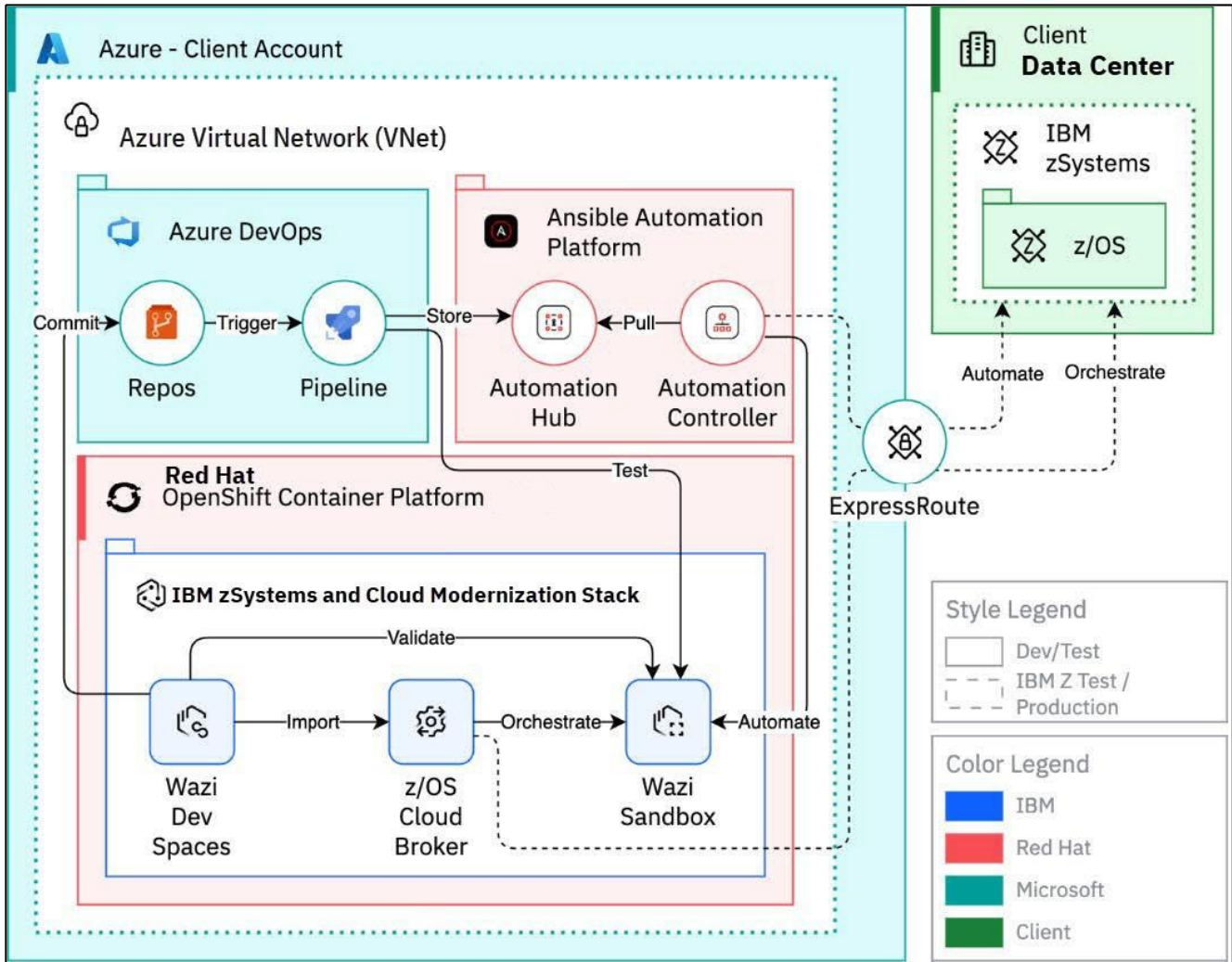


図 31 Microsoft Azure DevOps

IBM と共に着手する

IBM の専門家がお客様とともにイノベーションに取り組み、共創して、お客様の最も困難なビジネス上の課題を解決します。

- ▶ アプリケーション・モダナイゼーション・ジャーニーを加速します。
- ▶ 有意義かつスケーラブルなビジネス成果を達成します。

IBM と共創

IBM Consulting と IBM クライアント・エンジニアリングが、すべての業界を対象に有意義かつスケーラブルなビジネス成果を実現します。IBM は、多分野の深い知識を持つチームによる人間中心のアプローチで、お客様組織のニーズに合わせて価値の高いエクスペリエンスとソリューションを提供します。

お客様環境で実施するカスタム・デモであれ、価値を実証するための MVP であれ、IBM はお客様のアプリケーション・モダナイゼーション・ジャーニーの段階に合わせてご支援し、ニーズに対応します。IBM クライアント・エンジニアリングは、IBM のテクノロジーとメソドロジーを活用してお客様をご支援し、共創とイノベーションを促進します。

IBM が提供する支援

- ▶ **困難なビジネス・ユースケースを解決:** お客様の最も悩ましい課題に対してユーザー中心のソリューションを開発し、高いビジネス成果を実現します。
- ▶ **わずか数週間で価値を実証:** お客様と共にソリューション案を定義し、数日から数週間で価値を実証します。
- ▶ **企業の拡張を支えるイノベーション:** IBM はお客様企業の成長に合わせて反復的に共創し、お客様が希望するプラットフォームでエンタープライズ規模のスケーラビリティを安全に実現します。
- ▶ **グローバルの専門家と連携:** ビジネス・テクノロジー・リーダー、テクノロジー・エンジニア、ソリューション・アーキテクト、デザイナー、およびデータサイエンティストで構成されるアジャイル・チームの深い業界知識を活用できます。
- ▶ **迅速にモダナイズ:** IBM のハイブリッドクラウドと AI のテクノロジーで迅速に構築し、変化し続ける市場の先を見据えることができます。
- ▶ **実績のあるパターンを使用:** IBM の実績のあるテクノロジー、アクセラレーター、およびメソドロジーで、デジタル・トランスフォーメーションを促進します。

おわりに

デジタル・トランスフォーメーションを加速し、生産性とビジネスの俊敏性を向上して、メインフレームとハイブリッドクラウド環境におけるスキル・ギャップを埋めるためには、適切な基盤を構築する必要があります。これを実現するために、IBM は次のベスト・プラクティスを推奨します。

- ▶ ハイブリッドクラウド・アプローチでメインフレーム・アプリケーション・モダナイゼーションを実施する。
- ▶ 継続的なアプリケーション・モダナイゼーション・ジャーニーを推進する。
- ▶ 推奨のクラウド・リファレンス・アーキテクチャーを活用して IBM Cloud、Amazon Web Services (AWS)、Microsoft Azure を利用する。
- ▶ IBM の実績のある共創メソドロジーで迅速に開始する。

さまざまなエントリー・ポイントを確認し、お客様のニーズ、環境、課題、および現在のビジネス・プロセスに合わせて、お客様のビジネスに直接価値を提供できるエントリー・ポイントを選択します。

IBM と協力して、IBM が推奨するアプリケーション・モダナイゼーション・ジャーニーを進めることで、より迅速に、低コストかつ低リスクでお客様のメインフレーム・アプリケーションをモダナイズできます。本書で説明した、ベスト・プラクティスとそれを支えるテクノロジー、リファレンス・アーキテクチャー、アクセラレーターをはじめとするアプローチや戦略は、お客様のデジタル・トランスフォーメーションの加速、および生産性とビジネスの俊敏性の向上に役立ちます。

著者

本書は、ポキプシー拠点にある IBM Redbooks との連携のもと、世界各国からの専門家チームによって作成されました。

Skyla Loomis
Vice President, IBM® Z Software

Suman Gopinath
IBM, STSM & Chief Architect, DevOps for IBM Z Hybrid Cloud

Peter McCaffrey
IBM, Principle Product Management GTM Lead, IBM Z Software

Tim Brooks
IBM, IBM Z AIOps Sr. Product Manager

Juergen Holtz
IBM, STSM, IBM Z Management Software

Bryant Panyarachun
IBM, STSM - Ansible for z/OS Ecosystem Architect

Purvi Patel
IBM, AI on IBM zSystems

Mythili Venkatakrisnan
IBM US, Distinguished Engineer, IBM Z Financial Services Sector CTO

Yichong Yu
IBM, STSM, IBM Cloud for Financial Services

このプロジェクトにご協力いただいた方々に感謝いたします。

Lydia Parziale, Certified IT Specialist, PMI Certified Project Manager, IBM Redbooks® Project Leader
IBM Redbooks, Poughkeepsie Center

Will Burroughs, Visual Designer
IBM Hursley

Michelle Carey, Content Editor and Instructional Designer, IMS
IBM Silicon Valley Lab

Wade Wallace, Senior Content Specialist and Editing Team Lead, IBM Redbooks
IBM Austin

Carlos Alvarado, STSM - IMS Development
IBM US

Reid Copeland, STSM - Compilers and Performance
IBM US

Ivan Dovgan, Chief Architect - Cloud Enablement for z/OS Applications
IBM US

Surya Duggirala, IBM Cloud Platform Engineering Guild Leader
IBM US

Jin Leem, UX & Visual Designer
IBM US

Skyla Loomis, Vice President, IBM Z Software
IBM US

Christian Miemiec, Manager IBM IT Economics, Americas and WW
IBM US

Anthony Papageorgiou, Technical Strategy Lead for Integration on IBM Z | Master Inventor
IBM US

このプロジェクトのレビューにご尽力いただいた方々に感謝いたします。

DeWayne Hughes
Principal, Z Client Architect, Financial Services Market

Kyle Charlet
IBM Fellow, CTO, IBM Z Software

Joanie Gines
VP WW, Z Application & Platform Software Sales Global Sales

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料は、IBM から他の言語でも提供されている可能性があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

記載されているパフォーマンス・データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の今後の方針または意図に関する言明は、変更されたり予告なく撤回されたりすることがあり、目標を表明しているに過ぎません。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権の使用許諾：

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。したがって IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証したりすることはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、「Copyright and trademark information」(<https://www.ibm.com/legal/copytrade.shtml>) をご覧ください。

以下は、米国およびその他の国における International Business Machines Corporation の商標または登録商標です。

CICS®	IBM Research®	Redbooks®
Db2®	IBM Security®	Redbooks (ロゴ)  ®
GDPS®	IBM Watson®	zSystems®
IBM®	IBM Z®	z/OS®
IBM API Connect®	IBM zSystems®	z/VM®
IBM Cloud®	IBM z14®	z15®
IBM Cloud for Financial Services®	IBM z16™	z16™
IBM Cloud Pak®	Instana®	
IBM Consulting™	OMEGAMON®	

以下は、各社の商標または登録商標です。

Linux® は、世界中の商標の所有者である Linus Torvalds の独占的被許諾者である Linux Foundation からのサブライセンスに従って使用されます。

Zowe は Linux Foundation の商標です。

Microsoft および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Oracle および/またはその関連会社の商標または登録商標です。

Red Hat、Ansible、および OpenShift は、Red Hat, Inc. または子会社の米国およびその他の国における商標または登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。他の会社名、製品名またはサービス名等はそれぞれ各社の商標です。



REDP-5705-00

ISBN 0738461091

Printed in U.S.A.

Get connected

