



Java アプリケーションの モダナイゼーション・ ジャーニーの計画

成功のための実践ガイド

Java アプリケーションのモダナイゼーション・ジャーニーの計画

02

はじめに

ビジネス・アプリケーションをモダナイズする理由

絶え間ない変化には継続的な適応が必要

07

第1章

アプリケーションのモダナイゼーション・ジャーニーの計画

ニーズに合った方法を選択する

12

第2章

オープンソースのエキスパートとともにモダナイズする

成功するための基盤を Red Hat でデプロイする

17

第3章

成功事例を見る

実際のビジネス成果を確認する

19

まとめ

今すぐ始める

モダナイゼーションを今すぐ始めましょう

絶え間ない変化には継続的な適応が必要

今日のビジネスを取り巻く変化と不確実性

組織が成功するかどうかは、新たに導入した価値の高い製品やサービスを継続的に調整、変革し、顧客に提供する能力によって決まります。中核的なビジネス・アプリケーションや IT インフラストラクチャは、そうした目標を達成する上で重要な役割を果たすとともに、主要な差別化要因となる可能性があります。

同時に、既存のインフラストラクチャにおける時間や予算、トレーニングへの投資の価値を最大化する必要もあります。しかし多くの場合、本来イノベーションを実現するために使用できたはずのリソースを、そうしたシステムの管理に消費することになります。実際、調査によると、IT 組織は平均して予算の 64% を現行のインフラストラクチャとアプリケーションの運用に費やしており、イノベーションや成長に向けた取り組みにはわずか 36% しか使っていません。¹

アプリケーションのモダナイゼーションでリソースをイノベーションのために解放する

従来型のアプリケーションやレガシー・アプリケーションのサポートと管理には通常、膨大な時間、労力、資金が必要となります。そうしたアプリケーションはアーキテクチャの性質上、ビジネスや顧客の要求の変化に簡単に、または即座に適応することができません。また、ほとんどがサードパーティのシステムと完全には統合できないため、パートナーシップを通じた新しいサービスの追加が制限されます。

従来型のアプリケーションをモダナイズすることで、既存の投資からより多くのリターンを得るとともに、時間と予算を解放して戦略的なプロジェクト、機能の改善、ビジネスイニシアチブをサポートする新しいアプリケーションに割り当てることができます。クラウドネイティブなマイクロサービス・アーキテクチャとアプローチにより、短期的にはアプリケーションのアジリティとイノベーションを加速し、長期的にはより簡単に調整を行うために必要な効率性、柔軟性、スケーラビリティを実現できます。

しかし、こうしたテクノロジーの導入は非常に困難な作業になる場合があり、詳細な計画を策定して時間とリソースを配分する必要があります。そのため、大半の組織にとって、アプリケーションのモダナイゼーションは一度限りのことではなく、継続的なプロセスを意味します。

この e ブックでは、アプリケーション・ポートフォリオをモダナイズするメリットを説明するとともに、モダナイゼーションのプロセスの計画と実装のガイダンスを提供します。

先進的なハイブリッドクラウド・プラットフォームによるイノベーション

アプリケーションのモダナイゼーションとイノベーションを実現するためにハイブリッドクラウド・プラットフォームを導入した組織の成果：

20%

開発および DevOps チームの生産性の向上²

21%

IT インフラストラクチャ・チームの効率の向上²

636%

5 年間の投資対効果 (ROI)²

\$21.6M

年間売上高の増加²

¹ Flexera, 「Flexera state of tech spend report」、2021 年 1 月。

² IDC ホワイトペーパー (Red Hat 後援) 「Red Hat OpenShift のビジネス価値」、Document #US47539121、2021 年 3 月。

アプリケーションをモダナイズするメリット



アジリティ

市場の状況や顧客の要求の変化に効率的かつ即座に適應する

- ▶ 魅力的で対応が早く、価値の高いカスタマーエクスペリエンスを実現し、競争力を維持する
- ▶ 新たな収益機会を利用する



スピード

新しい革新的なアプリケーション、サービス、機能を迅速に提供する

- ▶ 変更までのリードタイムを短縮し、アプリケーションやサービスのデプロイメントの頻度を増やす
- ▶ 開発者の生産性を向上させ、ビジネス価値の提供に集中する



コスト

運用、デプロイメント、インフラストラクチャのコストと技術的負債を削減する

- ▶ アプリケーション (およびアプリケーションの個々のサービス) を迅速かつ効果的に、コスト効率の高い方法でスケールアップする
- ▶ IT リソースを最適化して複雑性とコストを削減する



セキュリティ

IT とアプリケーションのセキュリティを向上させ、機密データを保護する

- ▶ 新しいテクノロジー、統合、アプローチを使用してセキュリティ体制を評価、再調整する
- ▶ 自動化を適用してセキュリティポリシーを継続的に遵守する



信頼性

アプリケーションのレジリエンシー (回復力) とインフラストラクチャの信頼性を高める

- ▶ アプリケーションやサービスで障害が発生した場合の平均修復時間 (MTTR) を短縮する
- ▶ アプリケーション変更の成功率を高めて安定性を向上させる

モダナイゼーションを成功させるために必要なこと

アプリケーションをモダナイズするためには、新しいテクノロジーだけでは不十分です。モダナイゼーションの取り組みから最大限のメリットを得るには、テクノロジーとプラットフォームを運用プロセスと組織文化に統合する包括的なアプローチが必要になります。

文化

スタッフがアイデアをオープンに共有し、一緒に問題を解決できるようにする

人材は、あらゆる大規模な IT およびビジネスイニシアチブの中核にあります。基幹業務、ネットワーク、セキュリティ、運用、開発、インフラストラクチャの各グループを含むすべてのチームが、モダナイゼーションの取り組みに参加してオープンに協力する必要があります。

プロセス

コラボレーションとコミュニティ形成を促進するプロセスでイノベーションを引き起こす

プロセスは組織内のプロジェクトを開始から終了まで進めるためのものです。アプリケーションの構築、デプロイ、管理、導入に関して合意された明確なプロセスとベストプラクティスは、モダナイゼーションの取り組みにとって不可欠です。

テクノロジー

スケーラビリティ、スピード、柔軟性を提供するオープン・ハイブリッドクラウドの基盤を構築する

基盤となる IT プラットフォームは、アプリケーションを構築、実行、管理するための機能を提供します。複数の世代のアプリケーションをサポートする統合されたハイブリッドクラウド・プラットフォームにより、自分のペースでモダナイズするのに必要な柔軟性とスケーラビリティを確保できます。



モダナイゼーションの主要な概念とテクノロジー

アプリケーションのモダナイゼーションには、主要なテクノロジーとアプローチがいくつか組み込まれています。これには、モダナイゼーション・プロセスに不可欠なものもあれば、特定のシナリオでのみ必要なものもあります。

クラウドネイティブ・アプローチ

クラウドネイティブ・テクノロジーにより、組織はプライベートクラウド、パブリッククラウド、ハイブリッドクラウドなどの先進的で動的なクラウド環境で、スケーラブルなアプリケーションを開発および実行することができます。クラウドネイティブ・アプローチを組織全体に適用することで、ビジネスや市場の状況の変化に応じてアプリケーションやサービスをより迅速に開発、デプロイ、改善、スケーリングできます。

マイクロサービス

マイクロサービス・アーキテクチャは、アプリケーションを一連の機能に分割します。これらの機能的なマイクロサービスは相互に分離していますが、連携して元のモノリシック・アプリケーションと同じタスクを実行します。マイクロサービスは一般に軽量で、個別のスケーリングや複数のアプリケーション間での共有が可能です。マイクロサービス・アーキテクチャは、クラウドネイティブ・アプローチとの整合性が高く、アプリケーションのモダナイゼーションでよく採用されます。

DevSecOps

DevSecOps (または **DevOps**) は、高品質なアプリケーションの提供を高速化するための文化、自動化、プラットフォーム設計に対するアプローチです。これにより、ビジネスの価値およびアジリティを向上させることができます。チーム間のコラボレーション、動的運用、標準化された環境に重点を置くことで、開発とデプロイメントのプロセスを迅速化します。DevSecOps では、セキュリティ、開発、運用の各チームのスタッフが連携して、可視性、フィードバック、ベストプラクティス、知見を共有します。

イベント駆動型アーキテクチャ

イベント駆動型アーキテクチャは、アプリケーション設計のモデルです。要求ではなくイベントのキャプチャ、通信、処理、永続性を中心に構造化されています。イベント駆動型アーキテクチャでは結合が最小限になるため、先進的な分散型環境およびアーキテクチャで適切に動作します。開発者は任意のリアクティブ・プログラミング言語でイベント駆動型アプリケーションを構築できるため、より高い柔軟性と豊富な選択肢を享受できます。



ハイブリッドクラウド・アプリケーション・プラットフォーム

ハイブリッドクラウド・プラットフォームは、データセンター、エッジ、およびクラウド環境で、ツール、アプリケーション、モデルを一貫した方法で開発、デプロイ、管理するための統合ソフトウェア基盤を提供します。主な機能は次のとおりです。

- ▶ 従来型の仮想化、コンテナ化されたクラウドネイティブ・アプリケーションのサポート
- ▶ IT 制御によるセルフサービスでのリソースの提供
- ▶ さまざまなタイプのクラウド環境間でのアプリケーションとワークロードの可搬性
- ▶ 統合された DevOps 機能と運用管理機能
- ▶ アプリケーション開発ツール
- ▶ サードパーティのテクノロジー・ベンダーおよびオープンソース・コミュニティからの統合

包括的で適応性の高いハイブリッドクラウド・プラットフォームは、アプリケーション・モダナイゼーションに向けたほとんどのプロセスにとって不可欠です。このプラットフォームを使うことで、自分のペースでより効率的に変革を進めることができます。



先進的な開発フレームワーク

開発フレームワークは、**Java™** などのプログラミング言語を使用してアプリケーションを構築している開発者向けに、プラグブルかつ拡張性の高いコードフローパターンを提供します。開発者は反復可能なフレームワークをアプリケーションのプログラミングのベースとして使用することで、開発時間を短縮し、一貫性を向上させることができます。たとえば、**Quarkus** や **Spring** といった **Kubernetes** ネイティブの **Java** 開発フレームワークを使用すれば、**コンテナ**、**マイクロサービス**、**サーバーレス**などの新しいインフラストラクチャ・テクノロジーを組み込むことができます。



サービスメッシュ

サービスメッシュは、マイクロサービス間の通信を提供するとともに、レジリエンシー（回復力）、セキュリティ、可観測性、ルーティング制御、知見などの機能を追加します。また、アプリケーションの各部分が相互にデータを共有する方法を制御します。サービスメッシュは、透明性が高くスケーラブルな専用インフラストラクチャ・レイヤーです。開発者はサービスメッシュを使用することで、標準的な基盤サービスを再構築したり接続したりすることなく、価値あるアプリケーションや機能を構築することに集中できます。



サーバーレスモデル

サーバーレスは、開発者が基盤となるインフラストラクチャを管理することなく、アプリケーションを構築および実行できるようにするクラウドネイティブ開発モデルです。サーバーのプロビジョニング、保守、スケーリングといった定型作業はクラウドプロバイダーまたは IT 運用チームが行うため、開発者はコーディングに専念できます。デプロイ後は、サーバーレス・アプリケーションが需要やイベントトリガーに応じて自動的にスケールアップとスケールダウン、場合によってはゼロスケールを行い、インフラストラクチャで使用されていない貴重なリソースを解放できます。



CI/CD パイプライン

継続的インテグレーション/継続的デリバリー (CI/CD) パイプラインは、DevOps あるいはサイト信頼性エンジニアリング (SRE) のいずれかのアプローチを使用してソフトウェア提供を向上させます。CI/CD パイプラインは、自動化を適用してアプリケーション・ライフサイクルの管理プロセス（コードのコンパイル、単体テスト、コード分析など）を最適化します。コンテナ化されたクラウドネイティブ環境に移行する場合、多くの組織が、ハイブリッドクラウド・インフラストラクチャ間の提供を自動化する **GitOps** ワークフローを使用して CI/CD パイプラインを拡張します。



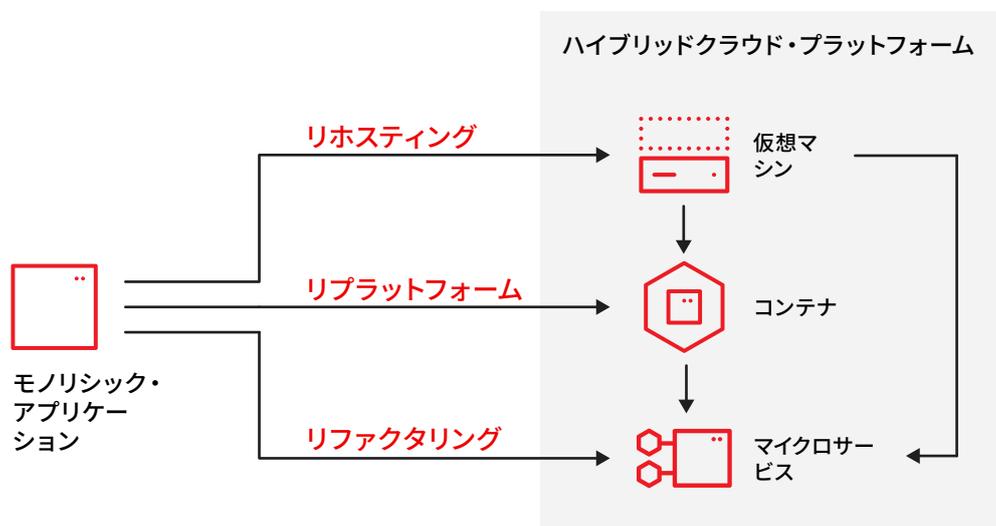
アプリケーションのモダナイゼーション・ジャーニーの計画

アプリケーションのニーズに合った方法を選ぶ

アプリケーションのモダナイゼーションを行う方法は主に 3 つあります。各方法にはテクノロジー、プロセス、文化、時間に関する固有の要件があり、それぞれメリットも異なりますが、いずれも組織がデジタル世界でより効果的に運用を行うのに役立ちます。

すべてのアプリケーションを同じ方法でモダナイズする必要はありません。各アプリケーションの特性と、組織の現在および予期されるニーズに最適な方法を選択できます。また、今は最小限の変更をアプリケーションに加えておき、要件の変化に合わせてさらにモダナイズすることも可能です。

- 1 方法1:リホスティング**
アプリケーションをハイブリッドクラウド・プラットフォーム上の仮想マシンにデプロイします。
- 2 方法2:リプラットフォーム**
Kubernetes ベースのハイブリッドクラウド・プラットフォーム上で実行されているコンテナにアプリケーションをデプロイします。
- 3 方法3:リファクタリング**
クラウドネイティブ・マイクロサービスを使用してアプリケーションを再構築し、新しいテクノロジーを統合して、ハイブリッドクラウド・プラットフォームにデプロイします。



方法1: リホスティング

既存のアプリケーションを、ハイブリッドクラウド・プラットフォーム上で実行されている仮想マシンにそのままデプロイします。

リホスティングによるモダナイゼーションでは、従来のアプリケーションサーバー上で実行されているアプリケーションを、ハイブリッドクラウド・プラットフォーム上で実行されている仮想マシンにリフトアンドシフトします。モノリシック・アプリケーションはアプリケーションサーバー上では変更されず、既存のすべての統合や依存関係は維持されます。外部データおよび統合は、既存のプラットフォーム上に維持できます。

リホスティングは一般に所要時間が短く、移行コストを抑えることができる一方、他のモダナイゼーション方法に比べてメリットが少なくなります。しかし、リホスティングを使うことで、仮想化、コンテナ化されたクラウドネイティブ・アプリケーションを単一の貫したプラットフォームに統合し、将来クラウドネイティブの運用を行う準備をすることができます。

以下に、モノリシックな Java アプリケーションをリホスティングした場合の例を示します。



企業は、カスタムのレガシー・アプリケーションの

20%

を、モダナイゼーションの取り組みの一環としてリホスティングすることを検討しています。³



ヒント:

一部の従来型サーバーは仮想マシンで動作しない場合があります。そのため、アプリケーションを仮想マシンに移行する前に、先進的なランタイム環境に再デプロイする必要があります。ランタイム環境を変更する必要がある場合は、アプリケーションをリプラットフォームしてコンテナにデプロイする (方法 2) ことで、モダナイゼーションの取り組みを最適化することを検討してください。

方法 2: リプラットフォーム

ハイブリッドクラウド・プラットフォーム上のコンテナで実行されている先進的なランタイム環境にアプリケーションを再デプロイします。

リプラットフォームでは、ハイブリッドクラウド・プラットフォーム上のコンテナで実行されている先進的なランタイム環境にアプリケーションをリフト、変更、シフトします。一部のアプリケーションについては、リプラットフォームにあたって他のアプリケーションよりも多くの変更が必要になります。たとえば、基本的な Java アプリケーションの場合、ほとんど変更することなく、OpenJDK などのコンテナ化された Java ランタイムのメリットを受けることができます。一方、エンタープライズ・アプリケーションの場合、コンテナにデプロイする前に、先進的なランタイム環境 (Red Hat® JBoss® Enterprise Application Platform、IBM WebSphere Liberty、Red Hat JBoss Web Server など) に移行する必要があります。

この方法は一般にリホスティングより時間がかかりますが、メリットも多くなります。アプリケーションを単一のハイブリッドクラウド・プラットフォームに統合することで、運用を効率化し、セルフサービス機能を提供できます。また、リプラットフォームしたアプリケーションでは、ハイブリッドクラウド・プラットフォームのすべてのネイティブ機能を利用できます。

以下に、モノリシックな Java アプリケーションをリプラットフォームした場合の例を示します。



企業は、カスタムのレガシー・アプリケーションの

18%

を、モダナイゼーションの取り組みの一環としてリプラットフォームすることを検討しています。⁴

方法 3: リファクタリング

アプリケーションを、ハイブリッドクラウド・プラットフォーム上のサービスメッシュにデプロイされたマイクロサービスとして再構築します。

リファクタリングでは、アプリケーション・サービスを、ハイブリッドクラウド・プラットフォーム上のサービスメッシュにデプロイされたマイクロサービスとして再開発します。サービスは時間をかけて再構築し、機能を古いアプリケーション・アーキテクチャから新しいアーキテクチャへと徐々に移行できます。再開発プロセスでは、基盤テクノロジーをアップグレードし、人工知能と機械学習 (AI/ML)、分析、自動スケーリング、サーバーレス機能、イベント駆動型アーキテクチャなどの新しいクラウドネイティブ機能を追加することもできます。

リファクタリングは最も時間がかかりますが、メリットも最も多くなります。リファクタリングでは、リホスティングとリプラットフォームで得られるすべてのメリットを享受しながら、革新的な新しいテクノロジーを採用してビジネスのアジリティと価値を高めることができます。

以下に、モノリシックな Java アプリケーションをリファクタリングした場合の例を示します。



企業は、カスタムのレガシー・アプリケーションの

17%

を、モダナイゼーションの取り組みの一環としてリファクタリングすることを検討しています。⁵

プロセスの計画を始める

アプリケーション・ポートフォリオ全体を評価して、アプリケーションごとに最適な方法を判断しましょう。

1

特定

現在のプラットフォームから新しいハイブリッドクラウド・プラットフォームに各アプリケーションを移行するにあたって、関連する技術面および運用面のリスクを判断します。2つのプラットフォームの違いと、それらの特性がアプリケーションに与える影響を理解してください。アプリケーションを新しいプラットフォームで実行するために必要な変更を文書に記録します。

2

評価する

モダナイゼーションの各方法に関連するコスト、メリット、懸念事項を評価します。

コスト：

- ▶ 開発にかかる時間と労力
- ▶ 継続的な運用のコスト
- ▶ プラットフォームのライセンスとインフラストラクチャのコスト
- ▶ ビジネス機会の損失

結果、以下のようなメリットが明らかになりました。

- ▶ 運用効率
- ▶ イノベーションの可能性
- ▶ 開発の生産性向上
- ▶ 新しいビジネス機会と収益機会

検討すべきその他の要因：

- ▶ アプリケーションの平均寿命
- ▶ 新しい機能の要件
- ▶ テクノロジースキルとコードの再利用
- ▶ プラットフォームのライセンス更新の時間枠

3

選択

目標、予算、アプリケーションの時間的制約に最適なモダナイゼーション方法を選択します。

4

検証

各移行が完了したら、アプリケーションが必要なビジネスロジックを正しく提供しているかどうか検証します。

アプリケーションにとってマイクロサービス・アーキテクチャが適切かどうかを見極める

多くの組織が、一部またはすべてのアプリケーションについて、マイクロサービス・アーキテクチャをデプロイする計画を立てています。各アプリケーションを評価して、アプリケーションを再構築することが理にかなっており、組織に価値をもたらすことを確認しましょう。たとえば、ユーザー数の少ない、またはスケーリングを行う必要がそれほどないアプリケーションの再構築にかかるコストは、それによって得られるメリットを上回る可能性があります。最終的な目標は、アプリケーションごとに最高の投資対効果をもたらすアーキテクチャと環境を選択することにあります。

オープンソースのエキスパートとともにモダナイズする

オープンソース・ソフトウェア、クラウドネイティブ・テクノロジー、Kubernetes のエキスパートである Red Hat は、お客様がアプリケーションをモダナイズしてアジャイルかつクラウドネイティブな IT 環境を構築し、進化するビジネス要求に対応できるよう支援します。アプリケーションのモダナイゼーション・ジャーニーのための、統一された完全なオープン・ハイブリッドクラウド基盤を提供します。Red Hat のソリューションにより、以下を実現できます。

- ▶ 現状を踏まえた上で、組織にとって適切なタイミングとペースで、必要な規模でアプリケーションをモダナイズする
- ▶ 従来型アプリケーションとモダナイズされたアプリケーションの両方をサポートするプラットフォームや製品を使用して、運用効率、レジリエンシー (回復力)、適応性を向上させる
- ▶ 同じプロセスを通じてすべてのアプリケーションを管理しながら、引き続き好みのランタイム、言語、開発ツールを使用する
- ▶ 標準化されたワークフロー、継続的インテグレーション、複数の環境のサポートにより、一貫性のある最適化された開発者エクスペリエンスを提供する

Red Hat は、お客様がモダナイゼーション・ジャーニーのどの段階にあっても、取り組みから最大限のメリットを得られるよう支援します。



アプリケーションのモダナイゼーション・ジャーニーを単純化する

Red Hat コンサルティングのエキスパートは、お客様と協力して、コストの削減、効率性の向上、開発の高速化に役立つアプリケーションのモダナイゼーション・ソリューションを評価、実装します。Red Hat のエキスパートが、お客様、そしてお客様のチームや組織がアプリケーションをすばやく構築し、組織全体で進化させるために必要なプラクティス、ツール、文化を生み出すお手伝いをします。

クラウドネイティブの成功を実現するために必要なスキルを身に付ける

Red Hat は、組織がスキルギャップを解消してビジネスの課題に対処するのに役立つ包括的なトレーニングコース・カリキュラムを提供しています。Red Hat ラーニングサブスクリプションには複数のレベルが用意されており、Red Hat トレーニングコースに無制限でアクセスできます。これには、オンラインの自習型コース、インストラクター主導型コース、クラウドベースのラボ、認定試験が含まれます。

モダナイゼーションのための一貫した基盤をデプロイする

Red Hat のプラットフォームは、アプリケーションのモダナイゼーションに必要な柔軟性、相互運用性、一貫性を提供します。



Red Hat OpenShift® はセキュリティ重視のハイブリッドクラウド・プラットフォームで、アプリケーションをあらゆる環境で確実に開発、デプロイ、管理できます。仮想化、コンテナ化されたクラウドネイティブ・ワークロードがサポートされているため、自分のペースでワークロードをモダナイズして移行しつつ、すべてのアプリケーションのライフサイクル管理を統合、最適化できます。このエンタープライズ・グレードのプラットフォームには、従来のモノリシック・アプリケーションのモダナイゼーション向けの主要コンポーネント（開発フレームワーク、CI/CD パイプライン、サーバーレス開発モデル、サービスメッシュなど）が含まれています。

Red Hat OpenShift は、開発者の生産性を向上させ、イノベーションを促進できるよう最適化されています。統合された自動化により、効率性が高く再利用可能なパイプラインを構築し、アプリケーションの開発、テスト、デプロイメントを高速化できます。開発者はセルフサービス機能を使用することで、IT チームの手を介さずに事前承認済みのリソースにすばやくアクセスできます。組み込みのツール、フレームワーク、ランタイムにより、すぐに使用を開始し、好みの言語やワークフローを引き続き使用できます。

Red Hat の大規模な**認定パートナーエコシステム**では、サードパーティの製品やサービスを利用して環境をカスタマイズできます。参加ベンダーの多くは、製品のインストールと管理を自動化および単純化する Red Hat OpenShift 認定 **Operator** も提供しています。

また、追加のセキュリティおよび管理コンポーネントを使って Red Hat OpenShift デプロイメントの価値を高めることもできます。**Red Hat Advanced Cluster Security for Kubernetes** では、Red Hat OpenShift 環境の脆弱性や脅威の経路についての詳細な情報を提供します。DevOps やセキュリティツールと統合することで、脅威を軽減し、アプリケーションの運用リスクを最小限に抑えるセキュリティポリシーを適用できます。**Red Hat Advanced Cluster Management for Kubernetes** は、Red Hat OpenShift クラスターの可視性と制御力を向上させます。マルチクラスター管理を統合し、ポリシーベースのガバナンスを提供するとともに、アプリケーションのライフサイクル管理を拡張します。

Red Hat OpenShift を使用してビジネス価値を得る

Red Hat OpenShift を活用する組織が実現しているメリット

20%

開発および DevOps チームの生産性の向上⁶

29%

アプリケーション開発ライフサイクルの迅速化⁶

21%

IT インフラストラクチャチームの効率の向上⁶



Quarkus は軽量のマイクロサービスおよびサーバーレス・アプリケーションを構築するための、Kubernetes ネイティブの Java フレームワークです。**Quarkus の Red Hat ビルド** は、Red Hat OpenShift に統合されており、メモリー使用量の少なさ、起動時間の短さ、一般的な Java 標準のサポート、強化された開発者エクスペリエンスを特徴としています。



Red Hat OpenShift Serverless はオープンソースの **Knative** プロジェクトをベースとしており、基盤となるインフラストラクチャからアプリケーション開発を抽象化します。トリガーされたときに起動し、需要に応じて動的にスケールリングして、使用後はゼロにスケールリングするイベント駆動型アプリケーションを構築、デプロイ、実行できます。



Red Hat OpenShift Pipelines は、オープンソース・プロジェクトの Tekton をベースとしたクラウドネイティブの CI/CD ソリューションです。Red Hat の開発者用ツールとの緊密な統合を通じて、個別のスケールリングが可能なパイプラインを、シンプルかつ繰り返し可能な手順ですばやく作成できます。



Red Hat OpenShift Service Mesh を使うと、マイクロサービスベースのアプリケーションを接続、管理、監視する方法を統一することができます。一連のオープンソース・プロジェクトで構成され、マイクロサービス間のトラフィックの管理、トレース、監視、分析を統合します。



Red Hat Application Foundations

Red Hat Application Foundations

Red Hat Application Foundations は、ソフトウェアの開発とモダナイゼーションのための包括的なコンポーネント一式を提供します。Application Foundations のテクノロジーにより、セキュリティを念頭に置き、ハイブリッドクラウド全体で大規模にアプリケーションを構築、デプロイ、運用できます。

アプリケーションのモダナイゼーション用の主要コンポーネントには、以下が含まれます。

- ▶ Red Hat の Quarkus ビルド
- ▶ Red Hat Single Sign-On
- ▶ Red Hat の OpenJDK ビルド
- ▶ Red Hat JBoss Enterprise Application Platform
- ▶ Red Hat Integration
- ▶ Red Hat JBoss Web Server
- ▶ アプリケーション移行ツールキット

自信を持って移行する

Red Hat は、モダナイゼーション・ジャーニーを計画してアプリケーションを Red Hat OpenShift に移行するのに役立つ**移行ツール**をいくつか提供しています。



仮想化移行ツールキット

仮想マシンを Red Hat OpenShift に大規模に移行する

仮想化移行ツールキット (MTV) は、VMware vSphere や Red Hat Virtualization から **Red Hat OpenShift Virtualization** に仮想マシンを移行するのに役立ちます。Red Hat OpenShift に組み込まれている Red Hat OpenShift Virtualization により、コンテナやクラウドネイティブのワークロードとともに仮想マシンを管理できます。MTV を使えば、大規模な移行を簡単に行うことができます。また、組み込みのオンサイト分析により、移行に伴う潜在的な問題を移行開始前に発見し、解決できます。

アプリケーションを Red Hat OpenShift にリホスティングする際に、MTV を使用して仮想マシンを移行できます。

アプリケーション移行ツールキット

アプリケーションをモダナイズしてクラウド対応環境に移行する

アプリケーション移行ツールキット (MTA) は、長年の経験に基づいて開発された、アプリケーションのモダナイゼーションと移行の大規模プロジェクトをサポートする一連のツールです。コード分析と移行を高速化し、作業量を見積もるとともに、クラウドおよびコンテナ環境へのアプリケーションの移行を支援します。

リプラットフォームやリファクタリングを行う際に、MTA を使用してアプリケーションを評価、分析できます。MTA により、アプリケーション・プラットフォームをアップグレードし、**Red Hat JBoss Enterprise Application Platform**、Spring、**Quarkus** の Red Hat ビルドなどのクラウドネイティブ・ランタイム環境に移行できます。

コンテナ移行ツールキット

Red Hat OpenShift クラスタ間でアプリケーションを移行する

コンテナ移行ツールキット (MTC) は、ステートレス・アプリケーションとステートフル・アプリケーションをさまざまな Kubernetes プラットフォームから Red Hat OpenShift 4 のクラスタに移行するのに役立ちます。MTC により、Kubernetes のリソース、永続ボリュームのデータ、内部のコンテナイメージを以下の場所から移行できます。

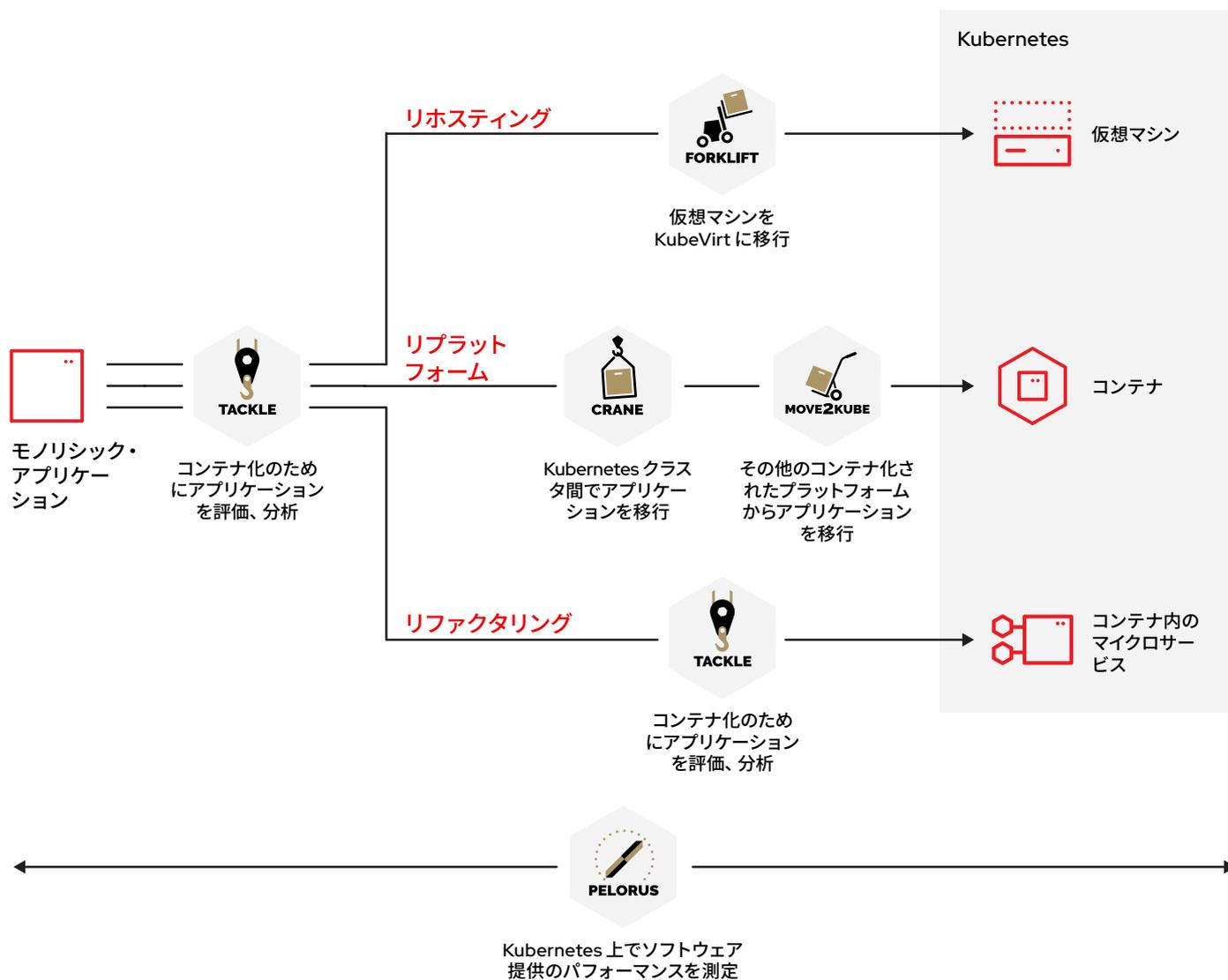
- ▶ Red Hat OpenShift 3 のクラスタ
- ▶ Red Hat Kubernetes 以外のプラットフォーム
- ▶ Red Hat OpenShift 4 の他のクラスタ

リプラットフォームやリファクタリングを行う際に、MTC を使用してクラスタ間でコンテナ化されたアプリケーションを移行できます。

Konveyor コミュニティに参加する

Red Hat と IBM は、組織がより簡単に Kubernetes ベースのアプリケーション・プラットフォームに移行できるよう、Konveyor コミュニティを作りました。**Konveyor** は、他の人々がオープンソースツールを使用してアプリケーションをモダナイズし、Kubernetes やクラウドネイティブ・テクノロジーにリホスティング、リプラットフォーム、リファクタリングするのをサポートするコミュニティです。

Konveyor は Red Hat と IBM の移行テクノロジー (MTV、MTC、MTA など) をベースとして、コミュニティが開発したプロジェクトを提供することで、Kubernetes への移行を単純化しています。また、Konveyor コミュニティでは定期交流会も開催しており、コミュニティのメンバーが学んだ内容についてディスカッションとデモを行っています。



成功事例を見る



水処理および管理テクノロジーの大手企業である DAB Pumps は、オープン・ハイブリッドクラウド・テクノロジーを使用して IT インフラストラクチャをモダナイズすることで、ビジネスの拡大をサポートし、プロセスとサービスの提供を高速化し、顧客エクスペリエンスを向上させたいと考えていました。この取り組みの重要な部分は、同社のさまざまなエンタープライズ・アプリケーション間の通信を改善するために、統合アーキテクチャをアップデートすることでした。

DAB Pumps は Red Hat OpenShift を使用して、モノリシックな環境からマイクロサービス・アーキテクチャへの移行を進めています。同社は Red Hat OpenShift への既存のミドルウェアの移行と、Red Hat OpenShift 上でのクラウドネイティブ・アプリケーションの開発を開始しました。また、**Red Hat Integration** テクノロジーを使用して、製品ライフサイクル管理 (PLM)、顧客関係管理 (CRM)、ビジネス・インテリジェンス (BI) などの重要なシステムに対するデータアクセスを統合し、高速化しました。

Red Hat のテクノロジーにより、DAB Pumps は統合されていないシステムを統合し、アプリケーションのロールアウト時間を 60% 短縮して、クラウド環境への移行を準備することができました。



インフラストラクチャ全体を一元管理



新しいアプリケーションの市場投入期間を 60% 短縮



組織間におけるリアルタイムでのデータの可用性を実現



「Red Hat とのコラボレーションにより、以前の 2 倍以上速く、より安全な方法で、適切に定義された基準に基づいて、新しいサービスやアップデートを市場に投入できるようになりました」

Stefano Dalla Grana 氏
DAB Pumps IT 統括責任者



Asiakastiето Group は、北歐地域で企業および消費者向けに革新的なデジタル情報サービスを提供する大手企業です。オープンバンキングへのシフトをサポートし、新しいデータプライバシーおよびセキュリティ要件に準拠するとともに、北歐の人々が抱える多額の負債を解決するため、同社はデータに基づいた信用評価ソリューションを開発することにしました。Asiakastiето は Red Hat OpenShift、Red Hat Integration、Quarkus を使用して、個人の返済能力のより正確な評価を通じて個人の負債と支払い不履行を削減するための Account Insight アプリケーションを開発しました。



デンマークの大手銀行向けの IT サービスプロバイダーである **Bankdata** は、高品質な IT ソリューションを開発、実装、実行する必要があります。同社は **Miracle** の協力のもと、Red Hat OpenShift を導入し、CI/CD プラットフォームを構築しました。リソースの効率を最大化してロールバックを高速化するため、Miracle と Bankdata のチームは現行の Spring Boot Java フレームワークのパフォーマンスと効率性を Quarkus と比較するテストを行いました。その結果、Quarkus ネイティブバージョンのテストアプリケーションでは、起動時間が短く、処理される呼び出しあたりのメモリー使用量が少なく、CPU 使用率も低いことがわかりました。



ルフトハンザ テクニックは、AVIATAR と呼ばれるデジタルプラットフォームを運用しています。このプラットフォームは、航空会社がデータを使用して整備を効率的に管理しスケジュールすることによって、遅延やキャンセルを回避できるように支援します。急成長と顧客からの要求の増加に対処するため、同社は Microsoft Azure 上の Red Hat OpenShift をベースとしたマイクロサービス・アーキテクチャに移行することにしました。AVIATAR のチームは、クラウドリソースの消費量を削減するため、Quarkus もデプロイしました。チームは現在、Quarkus を使用して新しいマイクロサービスを開発するとともに、既存の Spring Boot マイクロサービスを移行してパフォーマンスを強化し、リソースの使用量を最適化しています。



「オープンソースを利用することで、より広い範囲でイノベーションを加速できます。Red Hat は優れたセキュリティと価値あるサポートを提供してくれます」⁸

Reetta Sinelampi 氏
Asiakastiето 開発担当取締役

わずか

1秒未満

Quarkus ネイティブのアプリケーション・バージョンの起動時間

57%

Quarkus ネイティブ・バージョンのアプリケーションで、ロード時の呼び出しあたりのメモリー使用量を削減



「Quarkus を使用することで、[チームは] 可用性とサービスの応答時間を犠牲にすることなく、3 倍の密度でデプロイを実行できました」

Thorsten Pohl 氏
ルフトハンザ テクニック デジタル製品部門 AVIATAR
プロダクトオーナー自動化 & プラットフォーム・アーキテクト

⁸ Red Hat プレスリリース、「Asiakastiето、Red Hat でオープンバンキングのイノベーションの可能性を導く」、2020 年 5 月 27 日。

今すぐ始める

アプリケーション・ポートフォリオをモダナイズすることで、イノベーションを加速し、デジタル世界でより効果的に競争することができます。Red Hat は、取り組みを成功させるための製品、専門知識、サービスを提供します。自分のペースでモダナイズできると同時に、統一された完全な基盤により、すべてのアプリケーションの効率性、レジリエンス（回復力）、柔軟性を向上させることができます。

▶ [Red Hat OpenShift](#) を無料で試す

▶ [Red Hat OpenShift](#) についてさらに詳しく

モダナイゼーションの準備状況のセルフチェックを行う

Red Hat の無料のオンライン・セルフチェックで、アプリケーションのモダナイゼーションに向けた組織の準備状況や、モダナイゼーションを加速するためにできることを確認しましょう。



Red Hat コンサルティングを利用して、モダナイゼーション・ジャーニーをさらに前進させましょう。

Red Hat エキスパートは、新しいアプリケーションを構築するために必要なプラクティス、ツール、文化を生み出すお手伝いをします。

▶ [Red Hat コンサルティングの詳細はこちら](#)